A SOFTWARE PACKAGE FOR ANALYSIS OF GEOPHYSICAL MEASUREMENTS

John Frederick Arfman, Jr.

United States
Naval Postgraduate School

THESIS

A SOFTWARE PACKAGE FOR ANALYSIS
OF GEOPHYSICAL MEASUREMENTS

by

John Frederick Arfman, Jr.

Thesis Advisor: H. A. Titus

June 1971

Approved for public release; distribution unlimited.

A Software Package for Analysis of Geophysical Measurements

by

John Frederick Arfman, Jr.
Ensign, United States Navy
B.S., University of New Mexico, 1970

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1971

ABSTRACT

A software package for general analysis of real time series for geophysical data was developed. The package consists of FORTRAN IV callable subprograms that are assigned different tasks such as input, output (either printed or graphical), initial data analysis and filtering, spurious data rejection, FFT, and spectral analysis. A geophysical problem involving detection of unusual deep ocean pressures was analyzed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The types of real time series considered here are geophysical processes. Usually, time is the independent variable and such quantities as velocity and direction of flow, temperature, and pressure are the dependent variables. In the analysis and design of geophysical experiments, it is useful to have at one's disposal a set of software subroutines to aid in this analysis. Thus, a software package for general analysis of real time series geophysical data was developed. The package consists of FORTRAN IV callable subprograms that are assigned different tasks such as input, output (either printed or graphical), initial data analysis and filtering, spurious data rejection, FFT, spectral analysis, and so forth. A geophysical problem involving detection of unusual deep ocean pressures was analyzed.

II. A GEOPHYSICAL PROBLEM

A. STATEMENT OF THE PROBLEM

Throughout nature there exist many phenomenon that occur only as "unusual events." In the realm of oceanography one such unusual event is called a tsunami. The word tsunami comes from the Japanesse and means storm wave. It is known that the presence of a tsunami is indicated by a definite change in water pressure. Other information concerning the velocity and direction of the current, water temperature, salinity, and so forth, would also be useful in the analysis and understanding of these events. However, the infrequent rate at which tsunamis occur makes their observation difficult. In order to analyze and predict these events, methods were devised to detect and record them. The problem then is to logically design a detector that will automatically sense the presence of a tsunami and cause all the pertinent data to be recorded.

B. A MODEL

In order to determine the presence of an unusual event, a detector which receives digitized pressure data was designed [1]. Throughout the following discussion refer to Figure 1 for the logical design of the detector. The detector can be divided into six logically separate sections.

Data Input. Digital pressure sensors of varying degrees of sophistication and reliability are presently

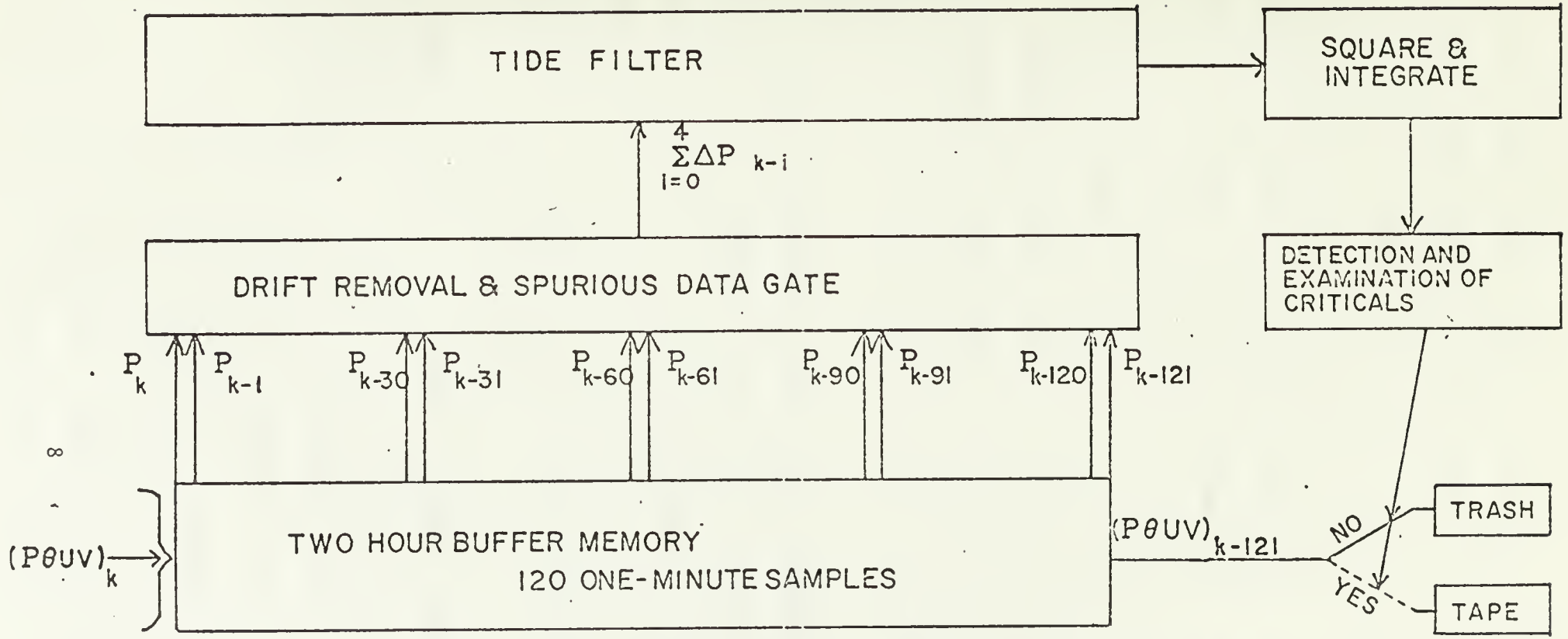

FIG. 1 BLOCK DIAGRAM OF UNUSUAL EVENT DETECTOR, IN THE EVENT OF A "CRITICAL" THE ENTIRE BUFFER, CONTAINING PRESSURE, TEMPERATURE AND VELOCITY COMPONENTS FOR THE LAST 120 ONE-MINUTE SAMPLES, IS WRITTEN ON TAPE (AND RETAINED IN BUFFER).

THE ABOVE SCHEMATIC REFERS TO DETECTION OF AN EVENT BAND OF PRESSURE; SIMILAR "CRITICALS" CAN BE TRIGGERED BY THE TEMPERATURE AND VELOCITY READINGS (Not shown).

available on the commercial market. Digital sensors generally count the number of cycles of a pressure sensitive oscillator over a given time interval. This summation in effect integrates the pressure signal. The integration process serves as a low pass filter. That is, high frequency pressure variations relative to the sampling interval tend to average out, thus contributing equally to each sample. This filtering of high frequency variations might prove undesirable in some applications. For this particular problem, however, almost all of the high frequency variations are naturally filtered out because of attenuation of pressure variations due to ocean depth.

It was determined that the frequency band in which tsunamis occur extends from ¼ to 30 cycles per hour. Nyquist frequency theory indicates that in order to detect a frequency of N cph, a sampling rate of at least 2N cph is required. Hence the sampling interval was determined to be one sample per minute.

Input Buffer. An input buffer receives the most recent pressure reading in digitized form. The actual hardware implementation of the input buffer is not part of the problem. Conceivably the input buffer could be anything from an acoustic delay line to a magnetic core.

The buffer is large enough to retain the most recent 121 consecutive samples. Hence, a two hour history of pressure data is available for analysis at any one time.

Figure 2(a) is a plot of typical pressure data over an 80 hour period.

First Differencing. The presence of a tsunami could be indicated by an abnormal change in pressure from one point in time to another. The phrase "change in pressure" indicates that the derivative should be computed. As a close approximation to the derivative, the positive first difference was determined.

Incorporated within the first differencer was a spurious data gate. Here each first difference was compared against a preset threshold value. If the threshold value was exceeded, then a sensor error (spurious error) was the probable cause since deep-ocean pressure changes are relatively small in amplitude. Since such an erroneous first difference might falsely indicate the presence of a tsunami, an excessive first difference was set equal to the previous first difference. The threshold value was determined after analysis of test data. First differencing, as computing the first derivative, removes the mean pressure value and suppresses drift. Figure 2(b) illustrates the effect of first differencing on the data of Figure 2(a).

Tide Filter. The purpose of the tide filter is to remove the effects of the diurnal and semi-diurnal tides. This was accomplished by forming a weighted sum from the sample data at ½ hour intervals. The coefficients were determined from analysis of the tidal spectrum. The tide filter output then is:

$$D_t = dP_t - 3.920392 \cdot dP_{t-30} + 5.841800 \cdot dP_{t-60}$$

$$- 3.920392 \cdot dP_{t-90} + dP_{t-120}$$

where dP_t is the first difference at time t [2].

Figure 2(c) shows the effect of the tide filter on the first differenced data of Figure 2(b). Note how the tide filter extended the effects of a spurious data value when there was no spurious data gating.

Rectification. As Figure 2(c) shows, the output of the tide filter presents an erratic picture of the changes in pressure. In order to observe any trends in this data, an integral over some fixed period of time needed to be computed. Any integration of the tide filtered data would only produce a result close to zero. This problem was overcome by taking the absolute value of the data in two ways; that is, the negative data values were made positive so that the data was no longer distributed about zero. The most straightforward method was to simply ignore the sign (i.e., rectification) of each number outputted from the tide filter. The second method was to square each value. Figure 2(d) shows the result of squaring the tide filter output.

Integration and Detection. The rectified or squared tide filtered data was then integrated. The method of integration was based on the solution of a differential equation. The general form of the integrating function is:

$$y_{t+1} = e^{-\alpha} (y_t + x_t) \quad , \quad t = 1,2,3,\ldots \text{ minutes}$$

and where $y_1 = 0$,

α = a time constant associated with the interval of integration,

x_t = the data to be integrated (rectified or squared output of tide filter).

The value of a was determined experimentally via the simulation ($\alpha = 1/60$). Figure 2(e) is a plot of the integrated tide filtered pressure data after the squaring process.

Dependent on the value of α, a critical threshold value had to be chosen such that whenever the integrated signal exceeded this value, a switch was closed causing the contents of the buffer to be recorded on a tape or otherwise be permanently recorded. At the end of a two hour period of normal input the buffer would once again be recorded. At this point the value of the integrated signal would be tested again. If the integrated signal was still greater than the threshold value, then the buffer would again be recorded after two hours; otherwise after two hours the testing is resumed as before. This technique guarantees that a minimum four hour record will be available for each event detected.

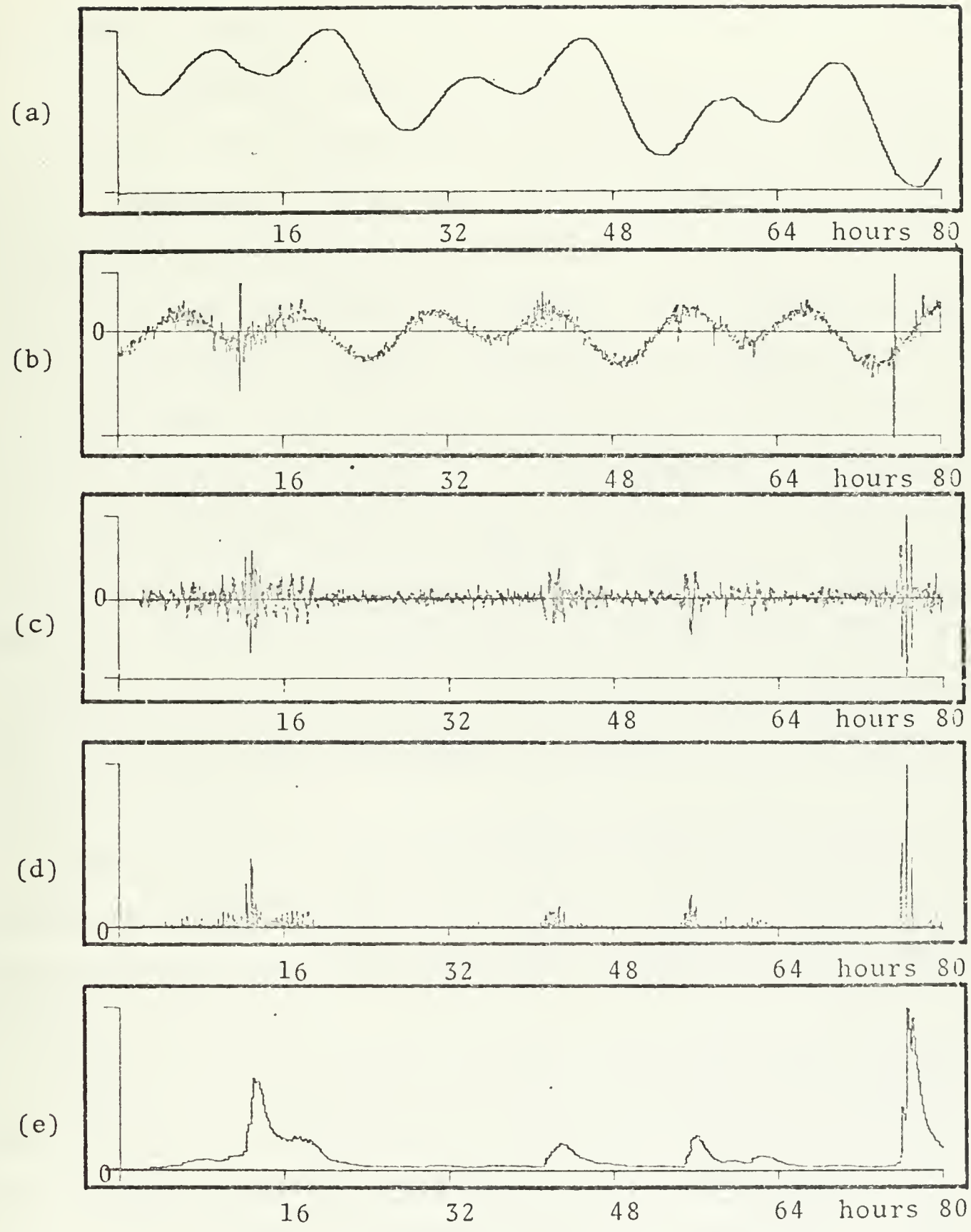

FIGURE 2. (a) Pressure, (b) First Differenced Pressure (spurious Data Gate Off), (c) Tide Filtered Pressure, (d) Squared Tide Filtered Data, (e) Integrated Signal Versus Time in Hours.

III. A SOFTWARE SIMULATION

A. GENERAL APPROACH

To be generally useful a software package must be able to simulate a real system on the level of logical components. That is, the basic information extracting procedures of the system must be available for graphical or statistical analysis during the simulation. Accordingly, the following set of eighteen FORTRAN callable subprograms was developed. The subprograms and their associated formal parameter are listed alphabetically in Table I. The subprogram UNIFRM is a function subprogram and XYAXIS is an entry in DRAW1. Otherwise all the subprograms listed in Table I are subroutines. The various subprograms are explained in the following sections according to their use in the unusual event problem. Each subprogram is listed in its entirety in the Appendix.

Each subroutine uses the variable FILE as its first formal parameter. The meaning and use of FILE will be given once here to avoid duplication in the sections which follow.

FILE is a real-valued variable identifying the current data file. This number is for user identification purposes only. FILE is usually set equal to 1.0 or the DO-loop index if the simulation is being performed iteratively. When a subroutine is called, a message is printed indicating that the subroutine has completed its task. The message is of the form:

SUBROUTINE	FORMAL PARAMETERS	PAGE
ANTIDE	(FILE, Y, TIDE, L)	26
CARDS	(FILE, Y, L)	16
DRAW1	(FILE, Y, L, YMAX, YMIN, LOG, XL, XI)	29
FFREAL	(FILE, A, B, N, ISN)	23
FFT	(FILE, A, B, N, ISN)	23
FIRSTD	(FILE, Y, DIFF, L, THRESH)	25
HARMON	(FILE, Y, YH, L, H1, H2)	24
INTGRL	(FILE X, Y, L, FTC)	27
OUTPUT	(FILE, Y, L)	28
PACK	(FILE, Y, L, Y2, L2, IDEN, RAMP, ISEED)	19
POWER	(FILE, Y, L, XL, XI)	25
RECSQR	(FILE, Y, Y2, L, IRS)	26
RESET	(FILE, Y, L, Y2, L2)	18
SCAN	(FILE, Y, L, YMEAN, YSTDEV, L2)	17
STATS	(FILE, Y, L, YMAX, YMIN, YMEAN, YSTDEV)	17
TAPE	(FILE, Y, L)	16
UNIFRM	(ISEED)	22
XYAXIS	(XAL, YAL)	29

TABLE I.

Subroutine Names, Formal Parameters, and Page References.

FILE NO. XX.X "message".

The digits XX.X indicate the current value of FILE. Execution of subroutine DRAW1 causes FILE to be incremented by 0.1.

B. DATA INPUT

CALL CARDS (FILE, Y, L)

CALL TAPE (FILE, Y, L)

Data input is accomplished by calling one of the above subroutines. The meaning of each of the formal parameters is:

FILE - see Section III.A.

Y - A real, singly dimensioned array into which the data is to be read.

L - An integer-valued variable equal to the size of array Y. In subroutine CARDS, L must equal the number of data elements to be read-in. In subroutine TAPE, L may equal the number of data elements to be read-in, or it may equal the maximum size of Y as defined in the DIMENSION statement of the main program. If an end-of-file mark is encountered on the tape before the L data elements have been read, the subroutine halts and returns the number of elements actually processed as the value of L.

These subroutines should be reloaded with the proper FORMAT statements inserted before the simulation is run. Currently the FORMAT is (10F7.0,F10.0).

C. INITIAL SCAN OF DATA

CALL STATS (FILE, Y, L, YMAX, YMIN, YMEAN, YSTDEV)

CALL SCAN (FILE, Y, L, YMEAN, YSTDEV, L2)

It is sometimes the case when geophysical data is being recorded that the initial start up of the sensing device intorudces spurious data which is recorded as valid data. Used in series, subroutines STATS and SCAN can determine the extent of any start up noise present in the data.

The meaning of each of the formal parameters is:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be analyzed. Neither subroutine alters the contents of array Y in any way.

L - An integer-valued variable equal to the size of array Y.

YMAX - A real-valued number returned by subroutine STATS equal to the maximum value encountered in array Y.

YMIN - A real-valued number returned by subroutine STATS equal to the minimum value encountered in array Y.

YMEAN - A real-valued number returned by subroutine STATS equal to the statistical mean of the data in array Y.

YSTDEV - A real-valued number returned by subroutine STATS equal to the standard deviation of the data in array Y.

L2 - An integer-valued number returned from subroutine SCAN equal to the number of noise dependent elements in array Y as determined from the following procedure.

1) set I=1, set ITEST=0.

2) If ABS(Y(I)-YMEAN)>2*STDEV then set ITEST=0, set L2=I;
otherwise, set ITEST=ITEST+1.

3) If ITEST > 100 then return; otherwise set I=I+1, go to step 2.

D. ADJUSTING THE SAMPLING INTERVAL

CALL RESET (FILE, Y, L, Y2, L2)

CALL PACK (FILE, Y, L, Y2, L2, IDEN, RAMP, ISEED)

UNIFRM (ISEED)

The sample size of data to be analyzed can be adjusted in two basic ways. First, the sampling interval may be shifted forward by any specified amount. For example, if preliminary analysis of the data indicates that the initial data elements are in error, then the data file can be reset so that the record begins later in the sequence, thereby skipping the bad data. This is accomplished by a call on subroutine RESET. The formal parameters of RESET are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be reset.

L - An integer-valued variable equal to the size of array Y.

Y2 - A real-valued, singly dimensioned array into which the reset elements of array Y are to be loaded. Arrays Y and Y2 may be the same physical array in which case the skipped data is overlayed and lost to further processing.

L2 - An integer-valued variable. On the initial call to RESET, L2 equals the number of data elements in array Y which are to be skipped. On return, L2 equals the size of array Y2.

The second method of adjusting the sampling interval involves increasing the density of the data points. This is accomplished by a call on subroutine PACK. The formal parameters of PACK are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be packed.

L - An integer-valued variable equal to the size of array Y.

Y2 - A real-valued, singly dimensioned array into which the expanded (packed) data of array Y is to be placed. Arrays Y and Y2 may be the same physical array.

L2 - An integer-valued variable. On the initial call to the subroutine, L2 must equal the maximum size of array Y2 as declared in the dimension statement of the main program. On return, L2 will equal the size of array Y2

as declared in the dimension statement of the main program. On return, L2 will equal the size of array Y2 if the subroutine completed successfully. The size of array Y2 can be determined from the formula (L-1)x IDEN+1.

IDEN - An integer-valued variable specifying the multiple of the current density that is desired. The value of IDEN in part determines the value of L2 that is returned to the main program. IDEN can also be interpreted to mean that IDEN-1 points are embedded between each of the original points in array Y.

RAMP - A real-valued variable equal to the maximum variability allowed when the embedded points are being computed. If RAMP = 0.0 then the embedded points are determined by linear interpolation between successive Y values; otherwise, the value of RAMP is multiplied by a uniformly random number between -½ and ½, and this value is added to the interpolated value.

ISEED - An integer-valued variable that serves as the random number seed for a random number generator. ISEED should be a six digit integer equal to ±3(mod 8).

Conceptually subroutine PACK embeds the data as shown in Figure 3.

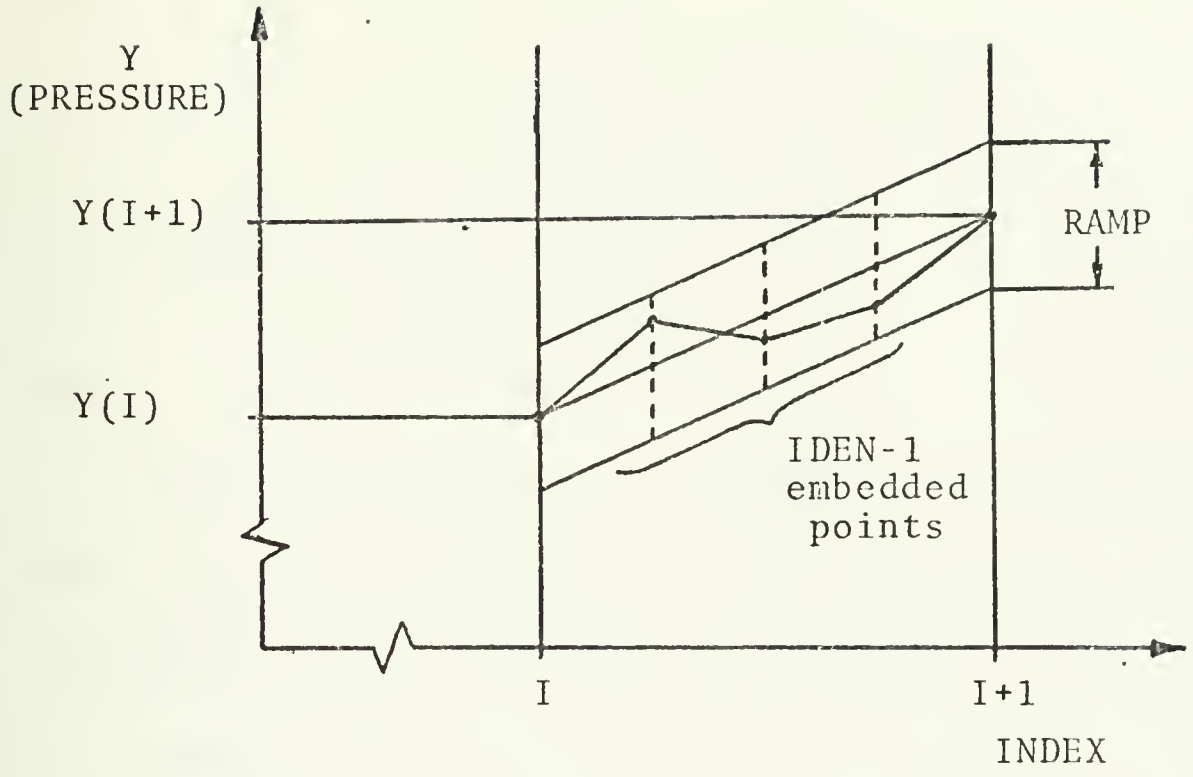

FIGURE 3.

Conceptual Operation of Subroutine PACK where RAMP > 0 and IDEN = 4.

Function subprogram UNIFRM returns a uniformly random number between -½ and ½. The use of UNIFRM is not restricted to subroutine PACK.

The original data used in the simulation was based on a four minute sampling interval. The detector, however, was designed to operate on one minute data. Hence, to obtain the proper sampling interval the following call was made on subroutine PACK.

```
CALL PACK (FILE, Y, L, Y, LMAX, 4, 0.0, 103941)
```

LMAX equals the maximum size of array Y as dimensioned in the main program. Note RAMP equals zero.

E. FOURIER ANALYSIS

```
CALL FFREAL   (FILE, A, B, N, ISN)
CALL FFT      (FILE, A, B, N, ISN)
CALL HARMON   (FILE, Y, YH, L, H1, H2)
CALL POWER    (FILE, Y, L)
```

Three areas of Fourier analysis are provided for in the software package. Subroutine FFREAL returns the Fourier cosine and sine coefficients of the data passed to it. Subroutine HARMON returns the approximation to the data passed to it on the basis of the H1 through H2 harmonics. Subroutine POWER returns the power spectrum of the data. Subroutine HARMON and POWER both call on FFREAL, and FFREAL in turn calls on subroutine FFT. Subroutines FFREAL and FFT are variations of Singleton's [3] algorithms for mixed radix fast Fourier transform.

The formal parameters of subroutine FFT are:

FILE - see Section III.A.

A, B - Real-valued, singly dimensioned arrays. Arrays A and B originally hold the real and imaginary parts of the data and return the real and imaginary Fourier coefficients.

N - An integer-valued variable equal to the number of real (or imaginary) data elements. In the context of this paper, N=L/2.

ISN - An integer-valued variable. The magnitude of ISN determined the step size of data in arrays A and B. The sign of ISN determines whether the complex or inverse transform will be performed.

The formal parameters for subroutine FFREAL are:

FILE - see Section III.A.

A, B, - Real-valued, singly dimensioned arrays. The original data values are alternately stored in arrays A and B. On return, arrays A and B hold the cosine and sine coefficients respectively, of the transform.

N - An integer-valued variable equal to the number of elements in array A. In the context of this paper, N=L/2.

ISN - An integer-valued variable. The magnitude of ISN determines the step size within arrays A and B. If the magnitude of ISN is 1, then

each consecutive element of the array is used in the computations. If the magnitude is 2, every other element is used. Hence, if the data is contained within a single array, say X, rather than two separate ones, the call to FFREAL would be

CALL FFREAL (X, X(2), N, 2).

If the sign of ISN is negative, then the inverse transform is computed.

The formal parameters of subroutine HARMON are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array originally containing the data to be analyzed. On return array Y contains the Fourier cosine and sine coefficients as returned from FFREAL.

YH - A real-valued, singly dimensioned array which on return holds the approximation based on the H1 and H2 harmonic of the data originally contained in array Y. This process involves a double summation and could be time consuming depending on the values of H1 and H2.

L - An integer-valued variable equal to the size of array Y (or array YH).

H1, H2 - Positive integer-valued variables indicating which harmonics, H1 through H2, inclusive, of array Y are to be isclated.

The formal parameters of subroutine POWER are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data which is to have its power spectrum computed. On return, the power spectrum coefficients are in the first N=L/2 elements of array Y.

L - An integer-valued variable equal to the size of array Y.

F. FIRST DIFFERENCING

CALL FIRSTD (FILE, Y, DIFF, L, THRESH)

Subroutine FIRSTD computes the positive first difference of the data in array Y, and returns the array of first differences in array DIFF. The first and last elements of DIFF are zero. Subroutine FIRSTD is so coded that arrays Y and DIFF may be the same physical array. By setting THRESH to any computed value, any first difference greater than THRESH will cause that first difference to be set equal to the previous first difference.

The formal parameters of subroutine FIRSTD are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be first differenced.

DIFF - A real-valued, singly dimensioned array which returns the positive first differences of the data in array Y.

L - An integer-valued variable equal to the size of array Y (or array DIFF).

THRESH - A real-valued variable against which each first difference is compared. If the current first difference is greater than THRESH (threshold value), then the first difference is set equal to the previous first difference.

G. TIDE FILTER

CALL ANTIDE (FILE, Y, TIDE, L)

Geophysical events usually occur in the domain of other less interesting phenomenon. In the case of the unusual event detector, one such phenomenon is the effect of the tides, both diurnal and semi-diurnal. Subroutine ANTIDE (anti-tide) filters out the combined effect of these tides.

The formal parameters of subroutine ANTIDE are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be tide filtered.

TIDE - A real-valued, singly dimensioned array into which the tide filtered data is placed. Arrays Y and TIDE may be the same physical array.

L - An integer-valued variable equal to the size of array Y (or array TIDE).

H. RECTIFYING THE SIGNAL

CALL RECSQR (FILE, Y, Y2, L, IRS)

Two methods of rectifying the data are provided. The

first involves taking the absolute value of each data value; while the second squares each data value.

The formal parameters of subroutine RECSQR (rectify-square) are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be rectified or squared.

Y2 - A real-valued, singly dimensioned array into which the rectified or squared data is to be placed. Arrays Y and Y2 may be the same physical array.

L - An integer-valued variable equal to the size of array Y (or array Y2).

IRS - An integer-valued variable equal to 1 or 2. If IRS = 1, then the data will be rectified by computing the absolute value of each element. If IRS = 2, then the data will be rectified by computing the square of each element.

I. INTEGRATION

CALL INTGRL (FILE, X, Y, L, FTC)

Integration is performed by multiplying the sum of previous data elements by a fading time constant (FTC); that is some positive number less than unity. The form of the integrating equation is:

$$Y(I) = EXP(-FTC) * (Y(I-1) + X(I))$$

where FTC is the fading time constant, and I is the index of the current variable.

The formal parameters of subroutine INTGRL are:

FILE - see Section III.A.

X - A real-valued, singly dimensioned array containing the data to be integrated.

Y - A real-valued, singly dimensioned array into which the integrated data is placed.

L - An integer-valued variable equal to the size of array X (or array Y).

FTC - A real-valued variable. A sample value of FTC could be 1/60.

J. OUTPUT

CALL OUTPUT (FILE, Y, L)

CALL DRAW1 (FILE, Y, L, YMAX, YMIN, LOG, XL, XI)

CALL XYAXIS (XAL, YAL)

Two means of output are provided. Subroutine OUTPUT prints out the first L elements of array Y. The page format of OUTPUT is 200 numbers with index numbers per page.

The formal parameters of subroutine OUTPUT are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be printed.

L - An integer-valued variable equal to the number of elements of array Y that are to be printed. L may equal the full size of array Y in which case a large quantity of paper is usually generated.

Subroutine DRAW1 plots the first L elements of array Y. The formal parameters of subroutine DRAW1 are:

FILE - see Section III.A.

Y - A real-valued, singly dimensioned array containing the data to be plotted. Subroutine DRAW1 does not alter the contents of array Y.

L - An integer-valued variable equal to the number of data points to be plotted.

YMAX, YMIN - Real-valued variables equal to the maximum and minimum values to be plotted. If YMAX $\leq$ YMIN, then the plot is autoscaled.

LOG - An integer-valued variable equal to 0 or 1. If LOG = 1, then the data in array Y is plotted on $\log_{10}$ scale. Otherwise, if LOG = 0 the data is plotted in the normal fashion.

XL, XI - Real-valued, singly dimensioned arrays containing literal constants. The maximum length string allowable is 40 characters. XL is the plot label which may be blank, and XI is the user's identification label.

The lengths of the plot axes are initialized to 20 inches for the x-axis and 8 inches for the y-axis. However, these dimensions may be changed by calling XYAXIS. The formal parameters of XYAXIS are:

XAL - A real-valued variable equal to the new length of the x-axis in inches.

YAL - A real-valued variable equal to the new length of the y-axis in inches. The value of YAL should be less than 9 inches.

Subroutine DRAW1 (and XYAXIS) make use of the standard plot package for the CALCOMP 765 plotter available on the IBM 360 at the Naval Postgraduate School.

IV. CONCLUSIONS

In the detector, the question arose as to what was the best value of the fading time constant used in the integration scheme. That is, given the form of the integrating equation and that integration is naturally a low pass filter, what value of the fading time constant would produce the most favorable signal-to-noise ratio? Three values of the time constant were chosen, and the integration was performed three times on the squared output of the tide filter with a different time constant in effect each time. The value of the constant used to produce Figure 2 was 1/60. The significance of this value is that because of the structure of the integrating equation, the contribution of a single data value after one hour's integration would have been reduced to one third of its original effect. This value of the time constant was used. The other two values of the time constant considered were 1/30 and 1/120. Table II shows the results of two signal-to-noise ratio computations using the three constants. The two hour "threshold" level to noise ratio was computed by finding a level such that the signal exceeded that level for exactly two hours.

The integrations were performed a second time but the rectified rather than the squared output of the tide filter was used. The ratios computed from these integrations are also given in Table II.

Signal-to-Noise Ratios for Unusual Event Detector	Tide Filter Output	Fading Time Constant=$e^{-\alpha}$ α=1/30.	α=1/60.	α=1/120.
Peak Signal to Noise Ratio	Rectified	8.33	5.86	4.57
	Squared	45.0	30.0	20.5
Two Hour "Threshold" to Noise Ratio	Rectified	4.00	4.44	3.64
	Squared	12.0	14.5	13.5

TABLE II.

SIGNAL-TO-NOISE RATIOS

From examination of Table II, it was determined that a value of 1/60 for the fading time constant gave the most desirable results when the integration was performed on the squared output of the tide filter. From inspection of the output statistics of the first differences, the standard deviation of the first differenced pressure data was approximately eight. Hence, a three sigma value of 25 was chosen as the threshold value of the spurious data gate in the first differencing subroutine. Figure 4 shows the effect of the spurious data gate on the detector output when this threshold limit was set at 25. Figure 4 also represents a fading time constant value of 1/60 applied to the integration of the squared output of the tide filter. (Compare Figures 2 and 4.)

The critical threshold value of the unusual event detector was determined to be three times that of the background signal of Figure 4(e). Hence, the three events detected in Figure 4(e) will cause the recorder to be activated.

The power spectra [4] of the tide filter and detector outputs are shown on Figures 5 and 6, respectively. In Figure 5 the null power readings coincide with even cph, and the peaks coincide with odd cph. This result agrees with the characteristics of the tide filter. The nodes of the power spectral plot of the detector output do not exactly agree with those of the tide filter because of the effect of the integration process.

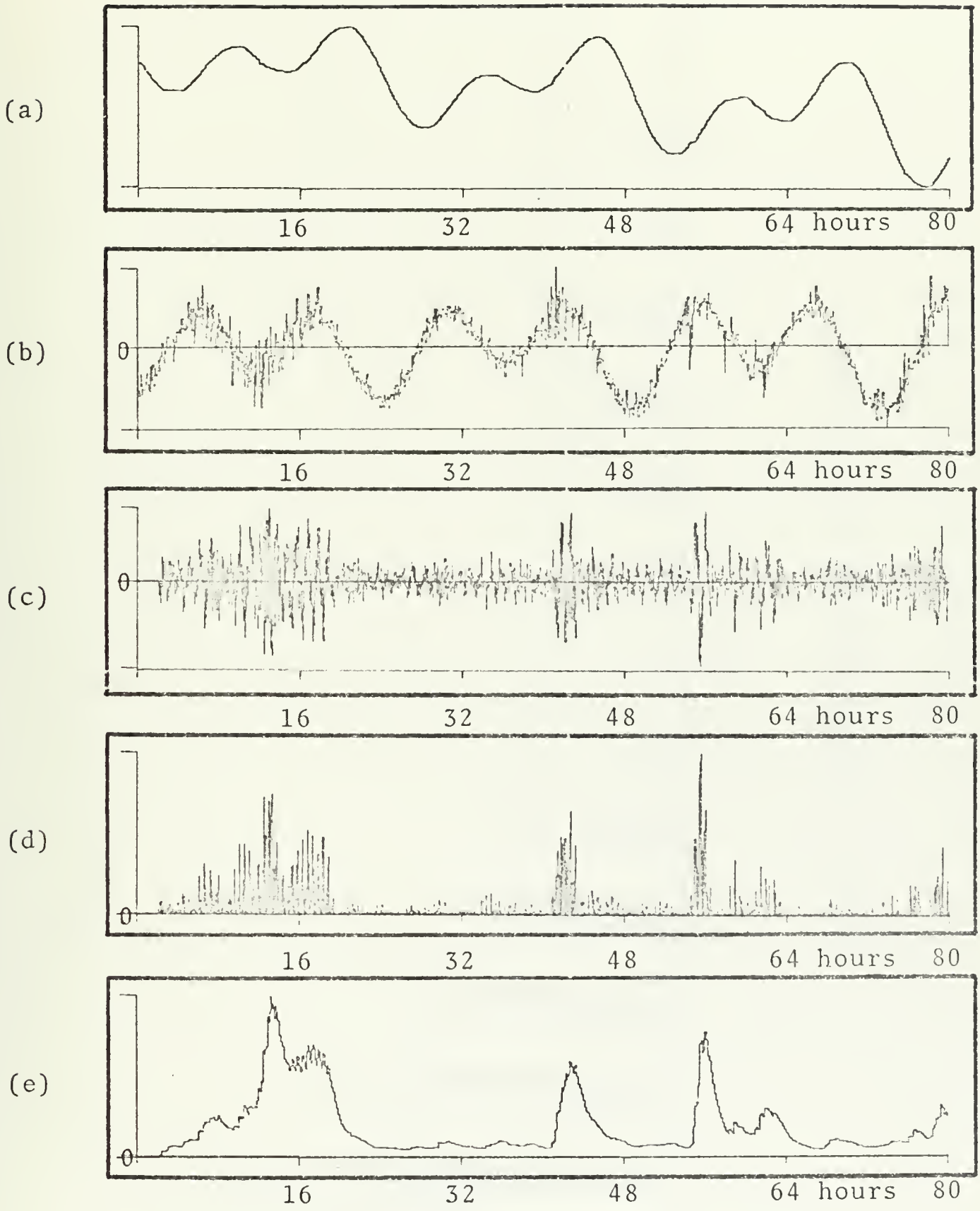

FIGURE 4. (a) Pressure, (b) First Differenced Pressure (Spurious Data Gate On), (c) Tide Filtered Pressure, (d) Squared Tide Filtered Pressure, (e) Integrated Pressure Versus Time in Hours.

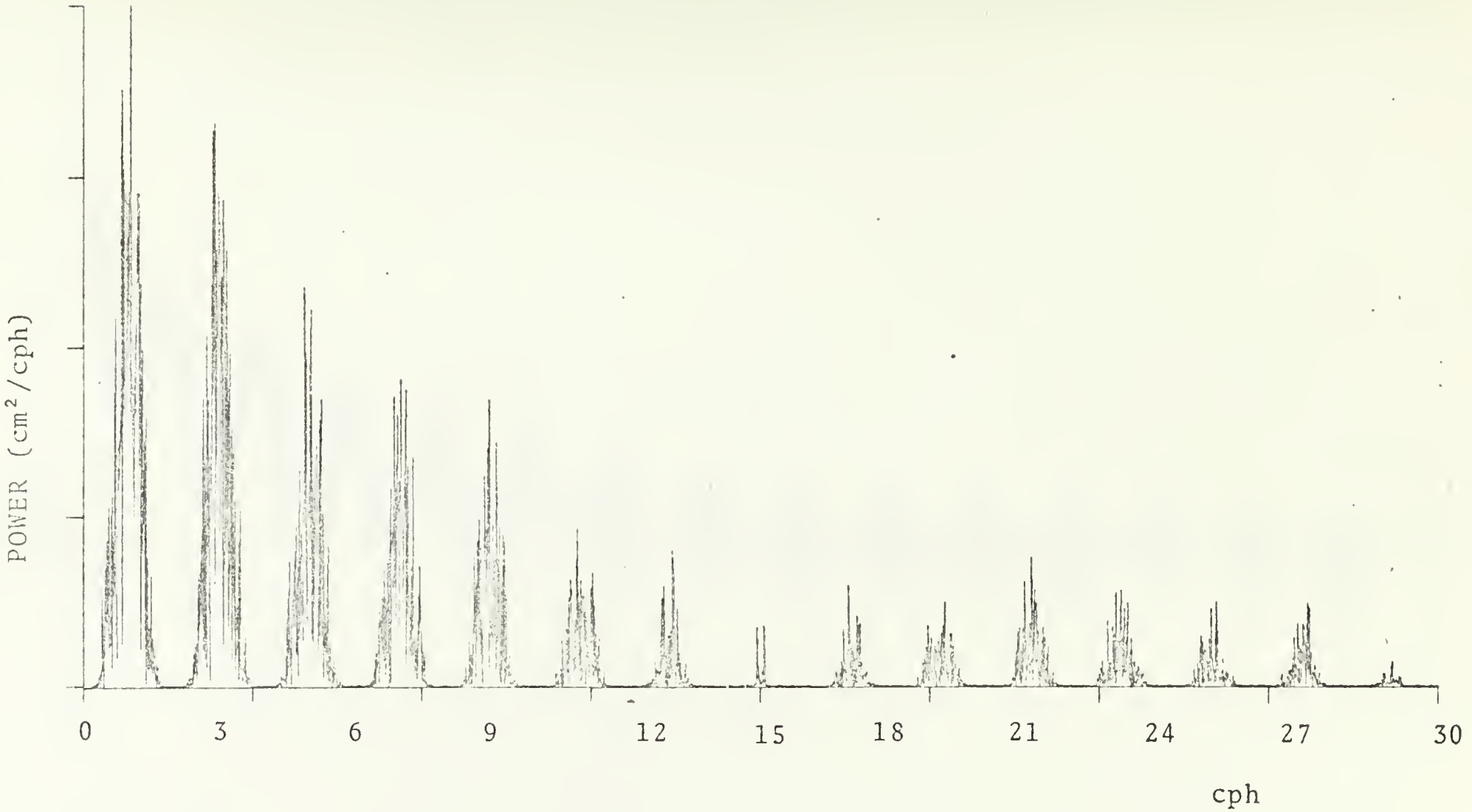

FIGURE 5. Power Spectrum of Tide Filter.

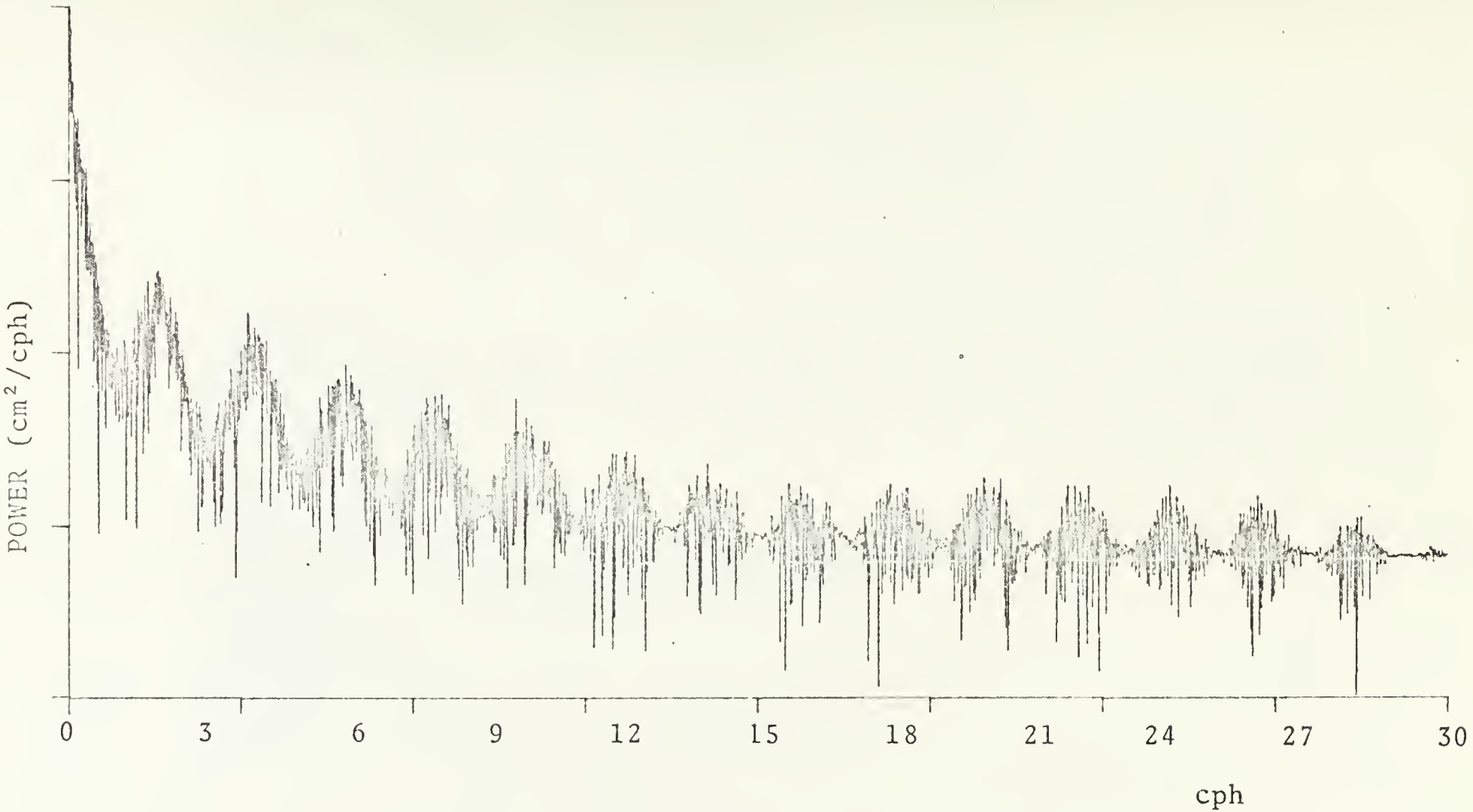

FIGURE 6. Power Spectrum of Detector Output.

```
      SUBROUTINE  ANTIDE  (FILE,Y,TIDE,L)
C
CC    PARAMETERS
CC      FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE TIDE
CC             DEPENDENT DATA WHICH IS TO BE FILTERED.
CC      TIDE = A REAL, SINGLY DIMENSIONED ARRAY INTO WHICH THE TIDE
CC             FILTERED DATA IS TO BE PLACED. ARRAYS Y AND TIDE MAY
CC             BE THE SAME PHYSICAL ARRAY.
CC      L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF ARRAY Y.
CC
CC    DESCRIPTION
CC      SUBROUTINE ANTIDE (ANTI-TIDE) REMOVES THE EFFECTS OF THE DIURNAL
CC    AND SEMI-DIURNAL TIDES ON THE DATA IN ARRAY Y AND RETURNS THE
CC    FILTERED DATA IN ARRAY TIDE. THE FIRST 120 ELEMENTS OF ARRAY TIDE
CC    ARE SET TO ZERO SINCE A 2 HOUR HISTORY OF DATA VALUES IS NECESSARY
CC    TO REMOVE THE EFFECTS OF THE TIDES. THE PROPER TIME SCALE IS
CC    ALSO MAINTAINED.
C
      REAL Y(L),TIDE(L)
      LM120=L-120
      DO 10 I=1,LM120
        J=L-I+1
        TIDE(J)=Y(J)-3.920392*Y(J-30)+5.8418*Y(J-60)
     .             -3.920392*Y(J-90)+Y(J-120)
   10 CONTINUE
      DO 20 I=1,120
        TIDE(I)=0.0
   20 CONTINUE
      WRITE(6,1) FILE
    1 FORMAT(1HO,'FILE ',F4.1,' TIDE FILTERED')
      RETURN
      END
```

```
      SUBROUTINE  CARDS  (FILE,Y,L)
C
C     PARAMETERS
C       FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
C              (FOR USER IDENTIFICATION PURPOSES ONLY.)
C       Y    = A REAL, SINGLY DIMENSIONED ARRAY INTO WHICH THE DATA IS
C              TO BE READ.
C       L    = AN INTEGER VARIABLE EQUAL TO THE NUMBER OF ELEMENTS TO
C              BE READ INTO ARRAY Y.
C
C     DESCRIPTION
C       SUBROUTINE CARDS READS FROM THE CARD READER (DEVICE 5), "L"
C     ELEMENTS. IN THE PRESENT CONFIGURATION, ONE LOGICAL RECORD
C     CONSISTS OF 10 ELEMENTS IN THE FORM 10F7.0,F10.0 (SEE FORMAT 1).
C
      REAL Y(L)
 1    FORMAT(10F7.0,F10.0)
 2    FORMAT(1H1,'READING FILE ',F4.1)
 3    FORMAT(1HO,'END OF FILE  ',F4.1,' - ',I5,' CARDS READ')
      WRITE(6,2) FILE
      NCARDS=0
      IFIRST=1
      ILAST=10
 10   READ(5,1,ERR=20,END=20)(Y(J),J=IFIRST,ILAST),CARD
      IFIRST=IFIRST+10
      ILAST =ILAST +10
      NCARDS=NCARDS+1
      IF(ILAST.GT.L) GO TO 20
        GO TO 10
 20   WRITE(6,3) FILE, NCARDS
      L=IFIRST-1
      RETURN
      END
```

```
      SUBROUTINE  DRAW1  (FILE,Y,L,YMAXV,YMINV,LOG,XL,XI)
C
CC    PARAMETERS
CC      FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC             FILE IS INCREMENTED BY .1 AFTER THE PLOT HAS BEEN DRAWN.
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE ORDINATE
CC             VALUES TO BE PLOTTED.
CC      L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF ARRAY Y.
CC      YMAXV, YMINV = THE MAXIMUN AND MINIMUM VALUES TO BE PLOTTED.
CC             IF YMAXV <= YMINV THEN THE PLOT IS AUTOSCALED.
CC      LOG  = AN INTEGER VARIABLE OF VALUE 0 OR 1. (LOG SCALE CODE.)
CC             IF LOG=1 THEN THE DATA IN ARRAY Y IS PLOTTED ON
CC             LOG10 SCALE.
CC      XL,XI= REAL ARRAYS CONTAINING LITERAL CONSTANTS. THE MAXIMUM
CC             CHARACTER STRING LENGTH ALLOWABLE IS 40 CHARACTERS.
CC              XL: LABEL DESCRIBING THE PLOT (MAY BE BLANK).
CC              XI: USER IDENTIFICATION (MAY NOT BE BLANK).
CC
CC    DESCRIPTION -
CC      SUBROUTINE DRAW1 PLOTS THE DATA CONTAINED IN ARRAY Y VERUS THE
CC    INDEX NUMBER OF THE DATA ON THE CALCOMP MODEL 765. THE LENGTH OF
CC    THE X AND Y AXES ARE INITIALLY 20 AND 8 INCHES RESPECTIVELY. THE
CC    DATA IS AUTOMATICALLY SCALED ACCORDING TO THE PASSED PARAMETERS
CC    YMAXV AND YMINV.
CC
CC    NOTE: THIS SUBROUTINE DOES NOT ALTER THE DATA PASSED TO IT.
C
      REAL XL(10),XI(10)
      REAL Y(L)
      DATA XLN/20./
      DATA YLN/8.0/
    1 FORMAT(1H0,'FILE ',F4.1,' PLOTTED')
      IF(L.LT.20) RETURN
      YMAX=YMAXV
      YMIN=YMINV
      GO TO 8
C
      ENTRY XYAXIS (XAL,YAL)
C
CC     PARAMETERS
CC      XAL  = THE LENGTH OF THE X AXIS IN INCHES.
CC      YAL  = THE LENGTH OF THE Y AXIS IN INCHES.
CC
CC    DESCRIPTION
CC      A CALL ON XYAXIS WITH THE PROPER PARAMETERS WILL CAUSE THE
C     LENGTHS OF THE X AND Y AXES TO BE CHANGED TO THE SPECIFIED SIZE.
```

```
C
      XLN=XAL
      YLN=YAL
      RETURN
 8    IF(YMAX.LE.YMIN) CALL STATS (FILE,Y,L,YMAX,YMIN,YMEAN,YSTDEV)
      IF(YMIN.LT.0.0) LOG=0
C  COMPUTE POINTS PER INCH AND RELATIVE ORIGINS (TRUNCATED TO INTEGER)
      IF(LOG.EQ.0) GO TO 10
      YPI=ALOG10(YMAX-YMIN+1.)/YLN
      YORG=ALOG10(YMIN+1.)
      GO TO 12
 10   YPI=AINT((YMAX-YMIN)/YLN)
      YORG=AINT(YMIN)
 12   XPI=FLOAT(L/IFIX(XLN))
      XORG=0.0
      YS=1./YPI
      XS=1./XPI
      EL=FLOAT(L)
C  INITIATE THE PLOT ROUTINE
      CALL PLOTS
      CALL SYMBOL (0.0,0.0,.28,XI,0.0,40)
      CALL SYMBOL (0.0,0.5,.28,XL,0.0,40)
      CALL NUMBER (0.0,1.5,.28,FILE,0.0,1)
      CALL SYMBOL (2.0,1.0,.21,'TOTAL POINTS PLOTTED =',0.0,22)
      CALL NUMBER (6.5,1.0,.21,EL,0.0,0)
      CALL SYMBOL (2.0,1.3,.21,'Y SCALE (UNITS/INCH) =',0.0,22)
      CALL NUMBER (6.5,1.3,.21,YPI,0.0,1)
      CALL SYMBOL (2.0,1.6,.21,'X SCALE (UNITS/INCH) =',0.0,22)
      CALL NUMBER (6.5,1.6,.21,XPI,0.0,1)
      CALL PLOT   (1.0,3.5,-3)
      CALL AXIS (YLN,0.,' ',+1,YLN,180.,YORG,YPI)
      IF(LOG.EQ.1) CALL SYMBOL (2.0,-0.6,.21,'LOG SCALE',180.,9)
      CALL AXIS (YLN,0.,' ',-1,XLN,90.0,XORG,XPI)
      IF(YMAX*YMIN.GT.0.0) GO TO 13
        S=YLN+YORG*YS
        CALL PLOT (S,-.125,3)
        CALL PLOT (S,XLN,2)
        CALL PLOT (S,-.125,2)
 13   IF(LOG.EQ.0) GO TO 14
      YP=YLN-ALOG10(Y(1)-YMIN+1.)*YS
      GO TO 15
 14   YP=YLN-(Y(1)-YORG)*YS
 15   XP=1.*XS
      CALL PLOT (YP,XP,3)
      DO 20 I=2,L
      IF(LOG.EQ.0) GO TO 16
      YP=YLN-ALOG10(Y(I)-YMIN+1.)*YS
      GO TO 17
```

```
16      YP=YLN-(Y(I)-YORG)*YS
17      XP=(FLOAT(I)-XORG)*XS
        CALL PLOT (YP,XP,2)
20      CONTINUE
        TOP=XLN+4.
        CALL PLOT (-1.,TOP,-3)
        CALL PLOTE
        WRITE(6,1)FILE
        FILE=FILE+.1
        RETURN
        END
```

```
      SUBROUTINE FFREAL (FILE,A,B,N,ISN)
C
CCC   PARAMETERS -
CCC       FILE= A REAL VARIABLE INDENTIFYING THE CURRENT DATA FILE.
CCC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CCC       A   = A REAL ARRAY CONTAINING THE DATA TO BE TRANSFORMED.
CCC             ON RETURN, ARRAY A CONTAINS THE COSINE AND SINE CO-
CCC               EFFICIENTS OF THE TRANSFORM FOR THE 1ST THROUGH THE
CCC               NTH HARMONIC.
CCC       B   = THE ADDRESS OF THE SECOND ELEMENT OF ARRAY A; I.E. A(2).
CCC       N   = AN INTEGER VARIABLE EQUAL TO 1/2 THE SIZE OF ARRAY A.
CCC             ON RETURN, N MAY BE SLIGHTLY SMALLER THAN THE VALUE
CCC               ORIGINALLY PASSED. (SEE SUBPROGRAM FFT.)
CCC       ISN = AN INTEGER OF VALUE +2 IF THE REAL TRANSFORM IS TO BE
CCC             PREFORMED; -2 IF THE INVERSE TRANSFORM IS DESIRED.
CCC
CCC   DESCRIPTION -
CCC     IF ISN=+2, SUBROUTINE FFREAL COMPUTES THE FOURIER TRANSFORM OF
CCC       2*N REAL DATA VALUES. THE COSINE AND SINE COEFFICIENTS
CCC       ALTERNATE AND ARE RETURNED IN ARRAY A. THE RESULTS ARE AUTO-
CCC       MATICALLY SCALED BY THE USUAL SCALE FACTOR OF 1/(2*N).
CCC     IF ISN=-2, SUBROUTINE FFREAL COMPUTES THE INVERSE FOURIER
CCC       TRANSFORM. IN THIS CASE THE SCALE FACTOR IS 1/2.
CCC     IN BOTH CASES, SUBROUTINE FFT IS CALLED TO PREFORM THE ACTUAL
CCC       TRANSFORM.
CCC
CCC   REFERENCE:
CCC    SINGLETON, R. C.,"ALGORITHM FOR MIXED RADIX FFT", IEEE TRANS-
CCC       ACTIONS ON AUDIO AND ELECTROACOUSTICS, VOL.AU-17, NO.2,
CCC       PP. 93-103, JUNE 1969.
C
      REAL A(1),B(1)
      REAL IM
    2 FORMAT(1HO,'FILE ',F4.1,' - FOURIER TRANSFORM')
      WRITE(6,2) FILE
      INC=IABS(ISN)
      SN=0.
      IF (ISN .LT. 0) GO TO 30
      CALL FFT (FILE,A,B,N,ISN)
    8 CN=1.0
      NK=N*INC+2
      NH=NK/2
      SD=2.0*ATAN(1.0)/FLOAT(N)
      CD=2.0*SIN(SD)**2
      SD=SIN(SD+SD)
      A(NK-1)=A(1)
      B(NK-1)=B(1)
      SC=1./(2.*N)
```

```
10    DO 20 J=1,NH,INC
      K=NK-J
      AA=A(J)+A(K)
      AB=A(J)-A(K)
      BA=B(J)+B(K)
      BB=B(J)-B(K)
      RE=CN*BA+SN*AB
      IM=SN*BA-CN*AB
      B(K)=(IM-BB)*SC
      B(J)=(IM+BB)*SC
      A(K)=(AA-RE)*SC
      A(J)=(AA+RE)*SC
      AA=CN-(CD*CN+SD*SN)
      SN=(SD*CN-CD*SN)+SN
      CN=0.5/(AA**2+SN**2)+0.5
      SN=CN*SN
20    CN=AA
      IF (ISN.GT.0) RETURN
      CALL FFT (FILE,A,B,N,ISN)
      RETURN
30    CN=-1.0
      NK=N*INC+2
      NH=NK/2
      SD=2.0*ATAN(1.0)/FLOAT(N)
      CD=2.0*SIN(SD)**2
      SD=SIN(SD+SD)
      SD=-SD
      SC=0.5
      GO TO 10
      END
```

```
      SUBROUTINE FFT (FILE,A,B,N,ISN)
C
CC    PARAMETERS -
CC      FILE = A REAL VARIABLE INDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      A    = A REAL ARRAY ORIGINALLY CONTAINING THE REAL COMPONENTS OF
CC             THE DATA. ON RETURN, ARRAY A CONTAINS THE REAL COMPONENTS
CC             OF THE FOURIER TRANSFORM; I.E. THE COSINE COEFFICIENTS.
CC      B    = A REAL ARRAY ORIGINALLY CONTAINING THE IMAGINARY COMPON-
CC             ENTS OF THE DATA. ON RETURN, ARRAY B CONTAINS THE IMAG-
CC             INARY COMPONENTS OF THE FOURIER TRANSFORM; I.E. THE SINE
CC             COEFFICIENTS.
CC      N    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF ARRAY A.
CC               (SIZE OF ARRAY A = SIZE OF ARRAY B.)
CC      ISN  : THE MAGNITUDE OF ISN DETERMINES THE STEP SIZE OF DATA IN
CC             THE ARRAYS A AND B. THE SIGN OF ISN DETERMINES WHETHER
CC             THE COMPLEX OR INVERSE TRANSFORM WILL BE PREFORMED.
CC
CC    DESCRIPTION -
CC      SUBPROGRAM FFT COMPUTES THE SINGLE-VARIATE COMPLEX FOURIER
CC    TRANSFORM COMPUTED IN PLACE USING A FIXED-RADIX FAST FOURIER
CC    TRANSFORM BY R. C. SINGLETON *.
CC      THE VALUE OF N MAY BE SLIGHTLY ALTERED BY THE SUBPROGRAM IN
CC    ORDER THAT ITS PRIME FACTORS SATISFY CERTAIN CONDITIONS.
CC      ARRAYS AT(MAXF), CK(MAXF), BT(MAXF), SK(MAXF), AND NP(MAXP)
CC    ARE USED FOR TEMPORARY STORAGE.  IF THE AVAILABLE STORAGE
CC    IS INSUFFICIENT, THE PROGRAM IS TERMINATED BY A STOP.
CC    MAXF MUST BE .GE. THE MAXIMUM PRIME FACTOR OF N.
CC    MAXP MUST BE .GT. THE NUMBER OF PRIME FACTORS OF N.
CC    IN ADDITION, IF THE SQUARE-FREE PORTION K OF N HAS TWO OR
CC    MORE PRIME FACTORS, THEN MAXP MUST BE .GE. K-1.
CC      ARRAY STORAGE IN NFAC FOR A MAXIMUM OF 13 PRIME FACTORS OF N.
CC      IF N HAS MORE THAN ONE SQUARE-FREE FACTOR, THE PRODUCT OF THE
CC    SQUARE-FREE FACTORS MUST BE .LE. 210.
CC    ARRAY STORAGE FOR MAXIMUM PRIME FACTOR OF 23.
CC    REFERENCE:
CC     SINGLETON, R. C.,"ALGORITHM FOR MIXED RADIX FFT", IEEE TRANS-
CC       ACTIONS ON AUDIO AND ELECTRCACOUSTICS, VOL.AU-17, NO.2,
CC       PP. 93-103, JUNE 1969.
CC
C
      REAL A(1),B(1)
      DIMENSION NFAC(13),NP(209)
```

```
      DIMENSION AT(23),CK(23),BT(23),SK(23)
      EQUIVALENCE (I,II)
C     SET NFAC TO ZERO
 1    DO 2 I=1,13
 2    NFAC(I)=0
C
      NTOT=N
      NSPAN=N
C  THE FOLLOWING TWO CONSTANTS SHOULD AGREE WITH THE ARRAY DIMENSIONS.
      MAXF=23
      MAXP=209
C
      IF (ISN .EQ. 0) GO TO 998
 4    INC=ISN
      C72=0.30901699437494742
      S72=0.95105651629515357
      S120=0.86602540378443865
      RAD=6.2831853071796
      IF(ISN .GE. 0) GO TO 10
      S72=-S72
      S120=-S120
      RAD=-RAD
      INC=-INC
10    NT=INC*NTOT
      KS=INC*NSPAN
      KSPAN=KS
      NN=NT-INC
      JC=KS/N
      RADF=RAD*FLOAT(JC)*0.5
      I=0
      JF=0
C
C  DETERMINE THE FACTORS OF N
C
      M=0
      K=N
      GO TO 20
15    M=M+1
      NFAC(M)=4
      K=K/16
20    IF(K-(K/16)*16 .EQ. 0) GO TO 15
      J=3
      JJ=9
      GO TO 30
25    M=M+1
      NFAC(M)=J
      K=K/JJ
30    IF(MOD(K,JJ) .EQ. 0) GO TO 25
```

```
      J=J+2
      JJ=J**2
      IF(JJ .LE. K) GO TO 30
      IF(K .GT. 4) GO TO 40
      KT=M
      NFAC(M+1)=K
      IF(K .NE. 1) M=M+1
      GO TO 80
   40 IF(K-(K/4)*4 .NE. 0) GO TO 50
      M=M+1
      NFAC(M)=2
      K=K/4
   50 KT=M
      J=2
   60 IF(MOD(K,J) .NE. 0) GO TO 70
      M=M+1
      NFAC(M)=J
      K=K/J
   70 J=((J+1)/2)*2+1
      IF(J .LE. K) GO TO 60
   80 IF(KT .EQ. 0) GO TO 92
      J=KT
   90 M=M+1
      NFAC(M)=NFAC(J)
      J=J-1
      IF(J .NE. 0) GO TO 90
 92   DO 94 III=1,13
      IF (NFAC(III) .GT. MAXF) GO TO 995
   94 CONTINUE
      NPROD = NFAC(1)
      DO 93 III=2,13
      IF (NFAC(III) .LT. 1) GO TO 96
      IIII = III -1
      DO 95 KKK = 1,IIII
      IF(NFAC(III) .EQ. NFAC(KKK)) GO TO 93
 95   CONTINUE
      NPROD = NPROD * NFAC(III)
   93 CONTINUE
 96   NPROD = NPROD - 1
      IF(NPROD .GT. MAXP) GO TO 995
      WRITE(6,98) FILE, N, NFAC
 98   FORMAT(1HO,'FILE ',F4.1,' IN FFT.',4X,I5,' TERMS'/
     .11X,'FACTORS:'/11X,13I5)
C
C   COMPUTE FOURIER TRANSFORM
C
  100 SD=RADF/FLOAT(KSPAN)
      CD=2.0*SIN(SD)**2
```

```
      SD=SIN(SD+SD)
      KK=1
      I=I+1
      IF(NFAC(I) .NE. 2) GO TO 400
C
C  TRANSFORM FOR FACTOR OF 2 (INCLUDING ROTATION FACTOR)
C
      KSPAN=KSPAN/2
      K1=KSPAN+2
  210 K2=KK+KSPAN
      AK=A(K2)
      BK=B(K2)
      A(K2)=A(KK)-AK
      B(K2)=B(KK)-BK
      A(KK)=A(KK)+AK
      B(KK)=B(KK)+BK
      KK=K2+KSPAN
      IF(KK .LE. NN) GO TO 210
      KK=KK-NN
      IF(KK .LE. JC) GO TO 210
      IF(KK .GT. KSPAN) GO TO 800
  220 C1=1.0-CD
      S1=SD
  230 K2=KK+KSPAN
      AK=A(KK)-A(K2)
      BK=B(KK)-B(K2)
      A(KK)=A(KK)+A(K2)
      B(KK)=B(KK)+B(K2)
      A(K2)=C1*AK-S1*BK
      B(K2)=S1*AK+C1*BK
      KK=K2+KSPAN
      IF(KK .LT. NT) GO TO 230
      K2=KK-NT
      C1=-C1
      KK=K1-K2
      IF(KK .GT. K2) GO TO 230
      AK=CD*C1+SD*S1
      S1=(SD*C1-CD*S1)+S1
      C1=C1-AK
      KK=KK+JC
      IF(KK .LT. K2) GO TO 230
      K1=K1+INC+INC
      KK=(K1-KSPAN)/2+JC
      IF(KK .LE. JC+JC) GO TO 220
      GO TO 100
C
C  TRANSFORM FOR FACTOR OF 3 (OPTIONAL CODE)
C
```

```
  320 K1=KK+KSPAN
      K2=K1+KSPAN
      AK=A(KK)
      BK=B(KK)
      AJ=A(K1)+A(K2)
      BJ=B(K1)+B(K2)
      A(KK)=AK+AJ
      B(KK)=BK+BJ
      AK=-0.5*AJ+AK
      BK=-0.5*BJ+BK
      AJ=(A(K1)-A(K2))*S120
      BJ=(B(K1)-B(K2))*S120
      A(K1)=AK-BJ
      B(K1)=BK+AJ
      A(K2)=AK+BJ
      B(K2)=BK-AJ
      KK=K2+KSPAN
      IF(KK .LT. NN) GO TO 320
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 320
      GO TO 700
C
C  TRANSFORM FOR FACTOR OF 4
C
  400 IF(NFAC(I) .NE. 4) GO TO 600
      KSPNN=KSPAN
      KSPAN=KSPAN/4
  410 C1=1.0
      S1=0
  420 K1=KK+KSPAN
      K2=K1+KSPAN
      K3=K2+KSPAN
      AKP=A(KK)+A(K2)
      AKM=A(KK)-A(K2)
      AJP=A(K1)+A(K3)
      AJM=A(K1)-A(K3)
      A(KK)=AKP+AJP
      AJP=AKP-AJP
      BKP=B(KK)+B(K2)
      BKM=B(KK)-B(K2)
      BJP=B(K1)+B(K3)
      BJM=B(K1)-B(K3)
      B(KK)=BKP+BJP
      BJP=BKP-BJP
      IF(ISN .LT. 0) GO TO 450
      AKP=AKM-BJM
      AKM=AKM+BJM
      BKP=BKM+AJM
```

```
      BKM=BKM-AJM
      IF(S1 .EQ. 0) GO TO 460
  430 A(K1)=AKP*C1-BKP*S1
      B(K1)=AKP*S1+BKP*C1
      A(K2)=AJP*C2-BJP*S2
      B(K2)=AJP*S2+BJP*C2
      A(K3)=AKM*C3-BKM*S3
      B(K3)=AKM*S3+BKM*C3
      KK=K3+KSPAN
      IF(KK .LE. NT) GO TO 420
  440 C2=CD*C1+SD*S1
      S1=(SD*C1-CD*S1)+S1
      C1=C1-C2
      C2=C1**2-S1**2
      S2=2.0*C1*S1
      C3=C2*C1-S2*S1
      S3=C2*S1+S2*C1
      KK=KK-NT+JC
      IF(KK .LE. KSPAN) GO TO 420
      KK=KK-KSPAN+INC
      IF(KK .LE. JC) GO TO 410
      IF(KSPAN .EQ. JC) GO TO 800
      GO TO 100
  450 AKP=AKM+BJM
      AKM=AKM-BJM
      BKP=BKM-AJM
      BKM=BKM+AJM
      IF(S1 .NE. 0) GO TO 430
  460 A(K1)=AKP
      B(K1)=BKP
      A(K2)=AJP
      B(K2)=BJP
      A(K3)=AKM
      B(K3)=BKM
      KK=K3+KSPAN
      IF(KK .LE. NT) GO TO 420
      GO TO 440
C
C  TRANSFORM FOR FACTOR OF 5 (OPTIONAL CODE)
C
  510 C2=C72**2-S72**2
      S2=2.0*C72*S72
  520 K1=KK+KSPAN
      K2=K1+KSPAN
      K3=K2+KSPAN
      K4=K3+KSPAN
      AKP=A(K1)+A(K4)
      AKM=A(K1)-A(K4)
```

```
      BKP=B(K1)+B(K4)
      BKM=B(K1)-B(K4)
      AJP=A(K2)+A(K3)
      AJM=A(K2)-A(K3)
      BJP=B(K2)+B(K3)
      BJM=B(K2)-B(K3)
      AA=A(KK)
      BB=B(KK)
      A(KK)=AA+AKP+AJP
      B(KK)=BB+BKP+BJP
      AK=AKP*C72+AJP*C2+AA
      BK=BKP*C72+BJP*C2+BB
      AJ=AKM*S72+AJM*S2
      BJ=BKM*S72+BJM*S2
      A(K1)=AK-BJ
      A(K4)=AK+BJ
      B(K1)=BK+AJ
      B(K4)=BK-AJ
      AK=AKP*C2+AJP*C72+AA
      BK=BKP*C2+BJP*C72+BB
      AJ=AKM*S2-AJM*S72
      BJ=BKM*S2-BJM*S72
      A(K2)=AK-BJ
      A(K3)=AK+BJ
      B(K2)=BK+AJ
      B(K3)=BK-AJ
      KK=K4+KSPAN
      IF(KK .LT. NN) GO TO 520
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 520
      GO TO 700
C
C   TRANSFORM FOR ODD FACTORS
C
  600 K=NFAC(I)
      KSPNN=KSPAN
      KSPAN=KSPAN/K
      IF(K .EQ. 3) GO TO 320
      IF(K .EQ. 5) GO TO 510
      IF(K .EQ. JF) GO TO 640
      JF=K
      S1=RAD/FLOAT(K)
      C1=COS(S1)
      S1=SIN(S1)
      IF (JF .GT. MAXF) GO TO 996
      CK(JF)=1.0
      SK(JF)=0.0
      J=1
```

```
630 CK(J)=CK(K)*C1+SK(K)*S1
    SK(J)=CK(K)*S1-SK(K)*C1
    K=K-1
    CK(K)=CK(J)
    SK(K)=-SK(J)
    J=J+1
    IF(J .LT. K) GO TO 630
640 K1=KK
    K2=KK+KSPNN
    AA=A(KK)
    BB=B(KK)
    AK=AA
    BK=BB
    J=1
    K1=K1+KSPAN
650 K2=K2-KSPAN
    J=J+1
    AT(J)=A(K1)+A(K2)
    AK=AT(J)+AK
    BT(J)=B(K1)+B(K2)
    BK=BT(J)+BK
    J=J+1
    AT(J)=A(K1)-A(K2)
    BT(J)=B(K1)-B(K2)
    K1=K1+KSPAN
    IF(K1 .LT. K2) GO TO 650
    A(KK)=AK
    B(KK)=BK
    K1=KK
    K2=KK+KSPNN
    J=1
660 K1=K1+KSPAN
    K2=K2-KSPAN
    JJ=J
    AK=AA
    BK=BB
    AJ=0.0
    BJ=0.0
    K=1
670 K=K+1
    AK=AT(K)*CK(JJ)+AK
    BK=BT(K)*CK(JJ)+BK
    K=K+1
    AJ=AT(K)*SK(JJ)+AJ
    BJ=BT(K)*SK(JJ)+BJ
    JJ=JJ+J
    IF(JJ .GT. JF) JJ=JJ-JF
    IF(K .LT. JF) GO TO 670
```

```
      K=JF-J
      A(K1)=AK-BJ
      B(K1)=BK+AJ
      A(K2)=AK+BJ
      B(K2)=BK-AJ
      J=J+1
      IF(J .LT. K) GO TO 660
      KK=KK+KSPNN
      IF(KK .LE. NN) GO TO 640
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 640
C
C   MULTIPLY BY ROTATION FACTOR (EXCEPT FOR FACTORS OF 2 AND 4)
C
  700 IF(I .EQ. M) GO TO 800
      KK=JC+1
  710 C2=1.0-CD
      S1=SD
  720 C1=C2
      S2=S1
      KK=KK+KSPAN
  730 AK=A(KK)
      A(KK)=C2*AK-S2*B(KK)
      B(KK)=S2*AK+C2*B(KK)
      KK=KK+KSPNN
      IF(KK .LE. NT) GO TO 730
      AK=S1*S2
      S2=S1*C2+C1*S2
      C2=C1*C2-AK
      KK=KK-NT+KSPAN
      IF(KK .LE. KSPNN) GO TO 730
      C2=C1-(CD*C1+SD*S1)
      S1=S1+(SD*C1-CD*S1)
      KK=KK-KSPNN+JC
      IF(KK .LE. KSPAN) GO TO 720
      KK=KK-KSPAN+JC+INC
      IF(KK .LE. JC+JC) GO TO 710
      GO TO 100
C
C   PERMUTE THE RESULTS TO NORMAL ORDER---DONE IN TWO STAGES
C   PERMUTATION FOR SQUARE FACTORS OF N
C
  800 NP(1)=KS
      IF(KT .EQ. 0) GO TO 890
      K=KT+KT+1
      IF(M .LT. K) K=K-1
      J=1
      NP(K+1)=JC
```

```
  810 NP(J+1)=NP(J)/NFAC(J)
      NP(K)=NP(K+1)*NFAC(J)
      J=J+1
      K=K-1
      IF(J .LT. K) GO TO 810
      K3=NP(K+1)
      KSPAN=NP(2)
      KK=JC+1
      K2=KSPAN+1
      J=1
C
CC   PERMUTATION FOR SINGLE-VARIATE TRANSFORM (OPTIONAL CODE)
C
  820 AK=A(KK)
      A(KK)=A(K2)
      A(K2)=AK
      BK=B(KK)
      B(KK)=B(K2)
      B(K2)=BK
      KK=KK+INC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 820
  830 K2=K2-NP(J)
      J=J+1
      K2=NP(J+1)+K2
      IF(K2 .GT. NP(J)) GO TO 830
      J=1
  840 IF(KK .LT. K2) GO TO 820
      KK=KK+INC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 840
      IF(KK .LT. KS) GO TO 830
      JC=K3
  890 IF(2*KT+1 .GE. M) RETURN
      KSPNN=NP(KT+1)
C
CC   PERMUTATION FOR SQUARE-FREE FACTORS OF N
C
      J=M-KT
      NFAC(J+1)=1
  900 NFAC(J)=NFAC(J)*NFAC(J+1)
      J=J-1
      IF(J .NE. KT) GO TO 900
      KT=KT+1
      NN=NFAC(KT)-1
      IF (NN .GT. MAXP) GO TO 997
      JJ=0
      J=0
```

```
      GO TO 906
  902 JJ=JJ-K2
      K2=KK
      K=K+1
      KK=NFAC(K)
  904 JJ=KK+JJ
      IF(JJ .GE. K2) GO TO 902
      NP(J)=JJ
  906 K2=NFAC(KT)
      K=KT+1
      KK=NFAC(K)
      J=J+1
      IF(J .LE. NN) GO TO 904
C
C   DETERMINE THE PERMUTATION CYCLES OF LENGTH GREATER THAN 1
C
      J=0
      GO TO 914
  910 K=KK
      KK=NP(K)
      NP(K)=-KK
      IF(KK .NE. J) GO TO 910
      K3=KK
  914 J=J+1
      KK=NP(J)
      IF(KK .LT. 0) GO TO 914
      IF(KK .NE. J) GO TO 910
      NP(J)=-J
      IF(J .NE. NN) GO TO 914
      MAXF=INC*MAXF
C
C   REORDER A AND B, FOLLOWING THE PERMUTATION CYCLES
C
      GO TO 950
  924 J=J-1
      IF(NP(J) .LT. 0) GO TO 924
      JJ=JC
  926 KSPAN=JJ
      IF(JJ .GT. MAXF) KSPAN=MAXF
      JJ=JJ-KSPAN
      K=NP(J)
      KK=JC*K+II+JJ
      K1=KK+KSPAN
      K2=0
  928 K2=K2+1
      AT(K2)=A(K1)
      BT(K2)=B(K1)
      K1=K1-INC
```

```
      IF(K1 .NE. KK) GO TO 928
  932 K1=KK+KSPAN
      K2=K1-JC*(K+NP(K))
      K=-NP(K)
  936 A(K1)=A(K2)
      B(K1)=B(K2)
      K1=K1-INC
      K2=K2-INC
      IF(K1 .NE. KK) GO TO 936
      KK=K2
      IF(K .NE. J) GO TO 932
      K1=KK+KSPAN
      K2=0
  940 K2=K2+1
      A(K1)=AT(K2)
      B(K1)=BT(K2)
      K1=K1-INC
      IF(K1 .NE. KK) GO TO 940
      IF(JJ .NE. 0) GO TO 926
      IF(J .NE. 1) GO TO 924
  950 J=K3+1
      NT=NT-KSPNN
      II=NT-INC+1
      IF(NT .GE. 0) GO TO 924
C
C
      RETURN
C
  995 N=N-1
      GO TO 1
C
  996 WRITE(6,1001)
 1001  FORMAT('  *** ERROR ***  MAX PRIME FACTOR EXCEEDED')
      STOP
C
  997 WRITE(6,1002)
 1002 FORMAT('  *** ERROR ***  MAX PRODUCT OF SQUARE FREE VALUES EXCEEDE
     .D')
      STOP
C
  998 WRITE(6,1003)
 1003 FORMAT('  *** ERROR *** ISN CANNOT BE ZERO')
      STOP
C
      END
```

```
      SUBROUTINE  FIRSTD (FILE,Y,DIFF,L,THRESH)
C
CC    PARAMETERS
CC      FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA TO
CC             BE FIRST DIFFERENCED.
CC      DIFF = A REAL, SINGLY DIMENSIONED ARRAY INTO WHICH THE POSITIVE
CC             FIRST DIFFERENCE OF THE DATA IN Y IS TO BE PLACED. ARRAYS
CC             Y AND DIFF MAY BE PHYSICALLY AND/OR LOGICALLY THE SAME
CC             ARRAYS.
CC      THRESH = A REAL VARIABLE AGAINST WHICH EACH FIRST DIFFERENCE IS
CC             CHECKED. IF THE CURRENT FIRST DIFFERENCE IS GREATER THAN
CC             THRESH (THRESHOLD VALUE), THEN THE FIRST DIFFERENCE
CC             IS SET EQUAL TO THE PREVIOUS FIRST DIFFERENCE.
CC
CC    DESCRIPTION
CC      SELF-EXPLANATORY
C
      REAL Y(L), DIFF(L)
      DIFF(1)=0.0
      LM1=L-1
      DO 10 I=2,LM1
        PFD=Y(I+1)-Y(I)
        IF(ABS(PFD).GT.THRESH) PFD=DIFF(I-1)
        DIFF(I)=PFD
   10 CONTINUE
      DIFF(L)=0.0
      WRITE(6,1) FILE
    1 FORMAT(1HO,'FILE ',F4.1,' FIRST DIFFERENCED')
      RETURN
      END
```

```
      SUBROUTINE  HARMON  (FILE,Y,YH,L,H1,H2)
C
CCC   PARAMETERS
CCC     FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CCC            (FOR USER IDENTIFICATION PURPOSES ONLY.)
CCC     Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA TO
CCC            BE ANALYIZED.
CCC     YH   = A REAL, SINGLY DIMENSIONED ARRAY WHICH ON RETURN TO THE
CCC            CALLING PROGRAM CONTAINS THE COMBINED EFFECT OF THE H1
CCC            THROUGH H2 HARMONICS.
CCC     L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF ARRAY Y.
CCC            SINCE ARRAY YH IS AN APPROXIMATION TO ARRAY Y, THE
CCC            SIZE OF ARRAY YH IS ALSO L.
CCC     H1,H2= POSITIVE INTEGER VARIABLES INDICATING WHICH HARMONICS,
CCC            H1 THROUGH H2 INCLUSIVE, OF ARRAY Y ARE TO BE COMPUTED.
CCC
CCC   DESCRIPTION
CCC     SUBROUTINE HARMON EXTRACTS THE EFFECTS OF THE SPECIFIED HAR-
CCC   MONICS, H1 THROUGH H2 INCLUSIVE. IN A SEMI-PERIODIC INTERVAL OF
CCC   L(=2N) EQUI-SPACED DATA ELEMENTS OVER A PERIOD T, L/2 HARMONICS
CCC   CAN BE EXTRACTED BY FOURIER ANALYSIS. THESE HARMONICS RANGE FROM
CCC   ZERO CYCLES PER PERIOD T (FUNDAMENTAL HARMONIC, MEAN SIGNAL, DC)
CCC   TO (L-1)/2 CYCLES PER PERIOD T (N-TH HARMONIC). SUBROUTINE HARMON
CCC   FORMS THE FOURIER EXPANSION OF THE DATA FROM THE H1 TO H2 HARMONIC
CCC   BY SUMMING THE FOURIER COEFFICIENTS RETURNED BY SUBROUTINE FFT.
CCC   (SEE LINE 16 BELOW)
CCC
CCC   NOTE: ARRAYS Y AND YH MUST BE PHYSICALLY DIFFERENT ARRAYS.
CCC         THE DATA IN ARRAY Y IS ALTERED.
C
      REAL Y(L), YH(L)
      INTEGER H1, H2, HS
      N=L/2
      CALL FFREAL  (FILE,Y,Y(2),N,+2)
      F1=1./FLOAT(L-1)
      IF(H1.GT.1) GO TO 10
        HS=3
        FTI=.5*Y(1)
      GO TO 11
   10   HS=H1
        FTI=0.0
   11 DO 20 I=1,L
        FT=FTI
        DO 18 J=HS,H2,2
          FT=FT+Y(J)*COS(3.14159*(J-1)*(I-1)*F1)
     .       +Y(J+1)*SIN(3.14159*(J-1)*(I-1)*F1)
```

```
18      CONTINUE
        YH(I)=FT
20    CONTINUE
      WRITE(6,1) FILE
1     FORMAT(1H0,'FILE ',F4.1,' - HARMONICS ISOLATED')
      RETURN
      END
```

```
      SUBROUTINE  INTGRL  (FILE,X,Y,L,FTC)
C
CC    PARAMETERS
CC      FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      X    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA TO
CC             BE INTEGRATED.
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY INTO WHICH THE INTE-
CC             GRATED RESULT OF ARRAY X IS PLACED. ARRAYS X AND Y MAY
CC             BE THE SAME PHYSICAL ARRAY.
CC      L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF ARRAY Y
CC           = THE SIZE OF ARRAY X.
CC      FTC  = FADING TIME CONSTANT. BY VARYING THE VALUE OF FTC THE
CC             LENGTH OF THE INTEGRATING INTERVAL CAN EFFECTIVELY BE
CC             ALTERED.
CC
CC    DESCRIPTION
CC      SUBROUTINE INTGRL PREFORMS INTEGRATION BY COMPUTING THE SUM OF
CC    ALL THE PREVIOUS ELEMENTS AND MULTIPLYING THE SUM BY A CONSTANT
CC    LESS THAN UNITY. THE INTEGRATING FUNCTION IS
CC
CCC         Y(I) = EXP(-FTC) * ( Y(I-1) + X(I) ) , I = 1,...,L.
C
      REAL X(L), Y(L)
      XINT=0.0
C
C   DELTA FUNCTION INTEGRATING TECHNIQUE
C
      EFTC=EXP(-FTC)
        DO 10 I=1,L
          XINT=EFTC*(XINT+X(I))
          Y(I)=XINT
 10     CONTINUE
        WRITE(6,1) FILE, FTC
 1      FORMAT(1HO,'FILE ',F4.1,' INTEGRATED.  FTC = ',F7.4)
      RETURN
      END
```

```
      SUBROUTINE OUTPUT (FILE,Y,L)
C
C     PARAMETERS -
C       FILE = A REAL VARIABLE INDENTIFYING THE CURRENT DATA FILE.
C              (FOR USER IDENTIFICATION PURPOSES ONLY.)
C       Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA
C              TO BE PRINTED.
C       L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF Y.
C
C     DESCRIPTION -
C       SUBROUTINE OUTPUT PRINTS THE DATA CONTAINED IN ARRAY Y. THE
C     OUTPUT FORMAT IS 4 COLUMNS PER PAGE, 50 NUMBERS PER COLUMN.
C     (200 DATA ELEMENTS PLUS INDEX NUMBERS PER PAGE.)
C
      REAL Y(L)
 1    FORMAT(1H1,45X,'D A T A    F I L E    N O .  ',F3.1,15X,'PAGE ',
     .I2,' OF ',I2,//,4(4X,'I',14X,'Y(I)',10X),/)
 2    FORMAT(4(1X,I6,7X,E13.6,5X,'|'))
      NCOL=(L-1)/50+1
      NPAGE=(NCOL-1)/4+1
      IPAGE=1
 8    WRITE(6,1) FILE, IPAGE, NPAGE
      IF(IPAGE.EQ.NPAGE) GO TO 20
      DO 10 I=1,50
      N1=(IPAGE-1)*200+I
      N2=N1+50
      N3=N1+100
      N4=N1+150
      WRITE(6,2)N1,Y(N1),N2,Y(N2),N3,Y(N3),N4,Y(N4)
 10   CONTINUE
      IPAGE=IPAGE+1
      GO TO 8
 20   LCOL=NCOL-(IPAGE-1)*4
      DO 30 I=1,50
      KCOL=LCOL+1
      N1=(IPAGE-1)*200+I
      N2=N1+50
      N3=N1+100
      N4=N1+150
      IF(N1+(LCOL-1)*50.GT.L) KCOL=KCOL-1
      GO TO(30,29,28,27,26),KCOL
 26   WRITE(6,2)N1,Y(N1),N2,Y(N2),N3,Y(N3),N4,Y(N4)
      GO TO 30
 27   WRITE(6,2)N1,Y(N1),N2,Y(N2),N3,Y(N3)
      GO TO 30
 28   WRITE(6,2)N1,Y(N1),N2,Y(N2)
      GO TO 30
 29   WRITE(6,2)N1,Y(N1)
 30   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PACK (FILE,Y,L,Y2,L2,IDEN,RAMP,ISEED)
C
CC    PARAMETERS -
CC      FILE = A REAL VARIABLE INDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA
CC             TO BE PACKED.
CC      L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF Y.
CC      Y2   = A REAL, SINGLY DIMENSIONED ARRAY INTO WHICH THE PACKED
CC             DATA IS PLACED.
CC      L2   = AN INTEGER VARIABLE EQUAL TO THE MAXIMUM SIZE OF
CC             ARRAY Y2.
CC      IDEN = AN INTEGER VARIABLE INDICATING WHAT MULTIPLE OF THE
CC             CURRENT DENSITY IS DESIRED.
CC      RAMP = A REAL VARIABLE WHICH INDICATES THE DEGREE OF VARIABILITY
CC             ALLOWED WHEN INSERTING THE EMBEDDED POINTS.
CC      ISEED= AN INTEGER VARIABLE WHICH IS THE INITIAL RANDOM NUMBER
CC             SEED FOR FUNCTION UNIFRM.
CC
CC    DESCRIPTION -
CC       SUBROUTINE PACK ACCEPTS ANY REAL ARRAY AND EMBEDS INTERPOLATED
CC    DATA POINTS BETWEEN THE ORIGINAL DATA POINTS. VARYING DEGREES OF
CC    RANDOMNESS CAN BE GIVEN TO EACH EMBEDDED VALUE BY SPECIFYING
CC    A NON-ZERO VALUE IN THE RAMP PARAMETER OF THE SUBROUTINE CALL.
CC    IF RAMP IS NON-ZERO, THEN RAMP IS MULTIPLIED BY A RANDOM NUMBER
CC    BETWEEN -.5 AND +.5, AND THIS VALUE IS ADDED TO THE EMBEDDED
CC    POINT.
CC       SUBROUTINE PACK IS SO WRITTEN THAT ARRAYS Y AND YP CAN BE THE
CC    SAME ARRAY; THEREBY SAVING STORAGE.
CC      FOR EXAMPLE; IF A DATA FILE ORIGINALLY OBTAINED FROM A 5 MINUTE
CC    SAMPLING INTERVAL NEEDS TO BE ALTERED TO A 1 MINUTE SAMPLING
CC    INTERVAL AND A VARIABILITY OF +/- 2 IS DESIRED; THEN THE CORRES-
CC    PONDING CALL ON PACK COULD BE:
CC
CC        CALL PACK (FILE,Y,L,Y2,L2,5,4.,103941)
CC      OR
CC        CALL PACK (FILE,Y,L,Y,L2,5,4.,103941)
CC          IF PACKED DATA TO OCCUPY SAME STORAGE AREA AS ORIGINAL.
C
      REAL Y(L),Y2(L2)
    1 FORMAT(1HO,'FILE ',F4.1,' PACKED')
      IF(IDEN.LE.0) RETURN
      LAST=(L-1)*IDEN+1
      IF(LAST.GT.L2) RETURN
```

```
      NIP=IDEN-1
      LM1=L-1
      DO 20 I=1,LM1
        IP=LAST-(I-1)*IDEN
        LI1=L-I+1
        RANGE=Y(LI1-1)-Y(LI1)
        Y2(IP)=Y(LI1)
        DO 10 J=1,NIP
        Y2(IP-J)=Y2(IP)+RANGE*FLOAT(J)/FLOAT(IDEN)
                       +RAMP*UNIFRM(ISEED)
10      CONTINUE
20    CONTINUE
      Y2(1)=Y(1)
      L2=LAST
      WRITE(6,1) FILE
      RETURN
      END
```

```
      SUBROUTINE POWER (FILE,Y,L)
C
C     PARAMETERS -
C       FILE = A REAL VARIABLE INDENTIFYING THE CURRENT DATA FILE.
C              (FOR USER IDENTIFICATION PURPOSES ONLY.)
C       Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA
C              WHOSE POWER SPECTRUM IS TO BE COMPUTED.
C       L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF Y.
C
C     DESCRIPTION -
C       SUBROUTINE POWER FINDS THE FOURIER TRANSFORM OF THE DATA IN
C     ARRAY Y. FROM THE TRANSFORM THE POWER SPECTRUM IS COMPUTED.
C     THE FOURIER COEFFICIENTS ARE COMPUTED BY SUBROUTINE FFREAL AND
C     SUBROUTINE FFT; ALGORITHMS WRITTEN BY R. C. SINGLETON OF THE
C     STANFORD RESEARCH INSTITUTE.
C
C     NOTE: THIS SUBROUTINE ALTERS THE DATA PASSED TO IT.
C
      REAL XL(10),XI(10)
      REAL Y(L)
      N=L/2
      CALL FFREAL (FILE,Y,Y(2),N,+2)
      DO 10 I=1,N
        Y(I) = Y(2*I-1)**2 + Y(2*I)**2
   10 CONTINUE
      WRITE(6,1) FILE
    1 FORMAT(1H0,'FILE ',F4.1,' - POWER SPECTRUM')
      RETURN
      END
```

```
      SUBROUTINE  RECSQR  (FILE,X,Y,L,IRS)
C
CC    PARAMETERS
CC      FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      X    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA TO
CC             BE SQUARED OR RECTIFIED.
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY WHICH RECEIVES THE
CC             SQUARED OR RECTIFIED DATA OF ARRAY X. ARRAYS X AND Y MAY
CC             BE PHYSICALLY THE SAME ARRAY.
CC      L    = AN INTEGER VARIABLE EQUAL TO 1 OR 2.
CC             IF IRS = 1 THEN THE ABSOLUTE VALUE OF THE ELEMENT OF
CC               ARRAY X IS PLACED IN ARRAY Y.
CC             IF IRS = 2 THEN THE SQUARED VALUE OF THE ELEMENT OF
CC               ARRAY X IS PLACED IN ARRAY Y.
CC
CC    DESCRIPTION
CC      SELF-EXPLANATORY
C
      REAL X(L),Y(L)
      IF(IRS.GE.2) GO TO 50
        DO 10 I=1,L
          Y(I)=ABS(X(I))
10      CONTINUE
        WRITE(6,1) FILE
1       FORMAT(1HC,'FILE ',F4.1,' RECTIFIED')
      RETURN
50      DO 60 I=1,L
          Y(I)=X(I)**2
60      CONTINUE
        WRITE(6,2) FILE
2       FORMAT(1HC,'FILE ',F4.1,' SQUARED')
      RETURN
      END
```

```
      SUBROUTINE  RESET   (FILE, Y, L, Y2, L2)
C
CC    PARAMETERS -
CC      FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA
CC             ELEMENTS.
CC      L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF ARRAY Y.
CC      Y2   = A REAL, SINGLY DIMENSIONED ARRAY INTO WHICH ARRAY Y WILL
CC             BE RESET. ARRAYS Y AND Y2 MAY BE PHYSICALLY AND/OR
CC             LOGICALLY THE SANE ARRAY.
CC      L2   = AN INTEGER VARIABLE WHOSE VALUE ON INPUT INDICATES WHICH
CC             ELEMENT OF ARRAY Y WILL BE THE FIRST ELEMENT OF ARRAY Y2.
CC             ON RETURN, L2 EQUALS THE SIZE OF ARRAY Y2.
CC
CC    DESCRIPTION -
CC      SELF-EXPLANATORY
C
      REAL Y(L), Y2(L)
    1 FORMAT(1H0,'FILE ',F4.1,' RESET BY ',I4)
      IF(L2.LT.0) RETURN
      LAST=L-L2
      DO 10 I=1,LAST
      Y2(I)=Y(I+L2)
   10 CONTINUE
      WRITE(6,1) FILE, L2
      L2=LAST
      RETURN
      END
```

```
      SUBROUTINE SCAN (FILE,Y,L,YMEAN,YSTDEV,L2)
C
C     PARAMETERS -
C       FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
C              (FOR USER IDENTIFICATION PURPOSES ONLY.)
C       Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA
C              TO BE FILTERED.
C       L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF Y.
C       YMEAN, YSTDEV = THE MEAN AND STANDARD DEVIATION OF THE DATA
C              CONTAINED IN ARRAY Y AS RETURNED BY SUBROUTINE STATS.
C       L2   = AN INTEGER VALUE WHOSE RETURNED VALUE IS EQUAL TO THE
C              NUMBER OF DATA ELEMENTS FILTERED FROM THE BEGINNING OF
C              THE ARRAY.
C
C     DESCRIPTION -
C       SUBROUTINE SCAN DETERMINES THE EXTENT OF ANY NOISE PRESENT
C     ON THE TAPE DUE TO INITIAL START UP OF THE RECORDING DEVICE.
C     THE FILTER CAN BE DESCRIBED AS BEING A TWO SIGMA FILTER. THAT IS,
C     FOR EACH ELEMENT IN THE INPUT ARRAY THE FOLLOWING TEST IS MADE:
C
C         IF ABS( Y(I) - YMEAN ) > 2 * YSTDEV )
C                  THEN INCREASE ITEST BY 1;
C                  OTHERWISE, SET ITEST = 0.
C
C         IF ITEST EVER REACHES AN ARBITRARY, PRESET VALUE (IN THIS
C     CASE 100) THE TEST IS TERMINATED.  L2 IS THE SET EQUAL TO THE
C     INDEX OF THE FIRST VALUE TO PASS THE TEST.
C
      REAL Y(L)
    1 FORMAT(1HO,'FILE ',F4.1,' SCANNED.   INITIAL ',I5,' SAMPLES NOISE
     .DEPENDENT.')
      ITEST=0
      SPREAD=2.*YSTDEV
      DO 20 I=1,L
        IF(ABS(Y(I)-YMEAN).GT.SPREAD) GO TO 10
        ITEST=ITEST+1
        IF(ITEST.GT.100) GO TO 30
        GO TO 20
   10   ITEST=0
        L2=I
   20 CONTINUE
   30 WRITE(6,1) FILE,L2
      RETURN
      END
```

```
      SUBROUTINE STATS (FILE,Y,L,YMAX,YMIN,YMEAN,YSTDEV)
C
CC    PARAMETERS -
CC      FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
CC             (FOR USER IDENTIFICATION PURPOSES ONLY.)
CC      Y    = A REAL, SINGLY DIMENSIONED ARRAY CONTAINING THE DATA
CC             WHOSE STATISTICS ARE TO BE COMPUTED.
CC      L    = AN INTEGER VARIABLE EQUAL TO THE SIZE OF Y.
CC      YMAX = A REAL VARIABLE WHOSE RETURNED VALUE IS EQUAL TO THE
CC             MAXIMUM VALUE OF Y.
CC      YMIN = A REAL VARIABLE WHOSE RETURNED VALUE IS EQUAL TO THE
CC             MINIMUM VALUE OF Y.
CC      YMEAN = A REAL VARIABLE WHOSE RETURNED VALUE IS EQUAL TO THE
CC             STATISTICAL MEAN OF THE DATA CONTAINED IN Y.
CC      YSTDEV = A REAL VARIABLE WHOSE RETURNED VALUE IS EQUAL TO THE
CC             STANDARD DEVIATION OF THE DATA CONTAINED IN Y.
CC
CC    DESCRIPTION -
CC      SUBROUTINE STATS COMPUTES THE REQUIRED STATISTICS (MAXIMUM
CC    AND MINIMUM VALUES, MEAN, AND STANDARD DEVIATION) OF ARRAY Y
CC    FOR USE IN SUBROUTINE DRAW1 AND SCAN.
C
      REAL Y(L)
    1 FORMAT(1HO,'FILE ',F4.1,' STATISTICS:'/
     .1HO,10X,'(BASED ON SAMPLE SIZE OF ',I5,')'/
     .1HO,10X,'MAXIMUM VALUE          ',E13.6/
     .1HO,10X,'MINIMUM VALUE          ',E13.6/
     .1HO,10X,'MEAN                   ',E13.6/
     .1HO,10X,'STANDARD DEVIATION     ',E13.6)
      YMAX=Y(1)
      YMIN=Y(1)
      YMEAN=Y(1)
      SSQUS=Y(1)**2
      DO 10 I=2,L
      IF(Y(I).GT.YMAX) YMAX=Y(I)
      IF(Y(I).LT.YMIN) YMIN=Y(I)
      YMEAN=YMEAN+Y(I)
      SSQUS=SSQUS+Y(I)**2
   10 CONTINUE
      YMEAN=YMEAN/FLOAT(L)
      YSTDEV=SQRT((SSQUS-FLOAT(L)*YMEAN*YMEAN)/FLOAT(L-1))
      WRITE(6,1) FILE, L, YMAX, YMIN, YMEAN, YSTDEV
      RETURN
      END
```

```
      SUBROUTINE TAPE (FILE,Y,L)
C
C     PARAMETERS -
C       FILE = A REAL VARIABLE IDENTIFYING THE CURRENT DATA FILE.
C              (FOR USER IDENTIFICATION PURPOSES ONLY.)
C       Y    = A REAL, SINGLY DIMENSIONED ARRAY WHOSE ELEMENTS ARE
C              ASSIGNED FROM THE INPUT TAPE.
C       L    = AN INTEGER VARIABLE INDICATING THE SIZE OF Y.
C
C     DESCRIPTION -
C       SUBROUTINE TAPE READS IN DATA FROM LOGICAL INPUT DEVICE 2 AND
C     ASSIGNS THIS DATA TO ARRAY Y IN LINEAR SEQUENCE. FORMAT STATEMENT
C     1 MUST REPRESENT THE FORMAT OF EACH LOGICAL INPUT RECORD. THE
C     INCREMENTS ON IFIRST AND ILAST AS WELL AS THE IMPLIED DO LOOP IN
C     STATEMENT LABELED 10, SHOULD BE COMPATABLE WITH FORMAT 1.
C       THE VALUE OF L ON INPUT TO THE SUBROUTINE SHOULD BE THE SIZE
C     OF THE PASSED ARRAY AS DIMENSIONED IN THE MAIN PROGRAM. THE
C     RETURNED VALUE OF L IS SET EQUAL TO THE ACTUAL SIZE OF THE ARRAY
C     AS READ IN FROM TAPE.
C       THE FOLLOWING 3 CARDS ARE AN EXAMPLE OF THE JOB CONTROL
C     LANGUAGE (JCL) CARDS REQUIRED TO READ FROM A TAPE DRIVE.
C     THESE ARE ONLY SAMPLE CARDS.
C
C    //GO.FT02F001  DD   UNIT=2400-1,VOL=SER=TAPENAME,LABEL=(1,NL),
C    //             DCB=(DEN=1,RECFM=F,BLKSIZE=80,TRTCH=ET),
C    //             DISP=(NEW,KEEP),DSN=DATA1
C
C
      REAL Y(L)
    1 FORMAT(10F7.0,F10.0)
    2 FORMAT(1H0,'READING FILE NO. ',F4.1,' - FIRST RECORD NO. =',F10.0)
    3 FORMAT(1H0,'DATA FILE NO. ',F4.1,' EMPTY - PROCEEDING TO NEXT FILE
   .')
    4 FORMAT(1H0,'ERROR IN RECORD NO. ',I5,' - RECORD SKIPPED')
    5 FORMAT(1H0,'END OF FILE NO.  ',F4.1,' - LAST RECORD NO.  =',F10.0,
   ./1H0,41X,I10,' RECORDS READ')
    6 FORMAT(1H1)
      WRITE(6,6)
      NREC=1
      GO TO 10
    8 WRITE(6,3) FILE
   10 READ(2,1,ERR=10,END=8) (Y(J),J=1,10),REC1
      WRITE(6,2) FILE, REC1
      IFIRST=11
      ILAST=20
   20 NREC=NREC+1
      READ(2,1,ERR=30,END=40)(Y(J),J=IFIRST,ILAST),RECNO
      IFIRST=IFIRST+10
```

```
      ILAST=ILAST+10
      IF (ILAST.GT.L) GO TO 40
      GO TO 20
30    WRITE(6,4) NREC
      GO TO 20
40    WRITE(6,5) FILE, RECNO, NREC
      L=IFIRST-1
      RETURN
      END
```

```
      FUNCTION UNIFRM (ISEED)
C
CC    COMPUTE A RANDOM NUMBER BETWEEN -.5 AND +.5
C
      ISEED=ISEED*103941
      UNIFRM=FLOAT(ISEED)*.2328306E-9
      RETURN
      END
```

COMPUTER OUTPUT

```
READING FILE  1.1
END OF FILE   1.1 -   143 CARDS READ
FILE  1.1 STATISTICS:
          (BASED ON SAMPLE SIZE OF  1430)
          MAXIMUM VALUE          0.242438E 06
          MINIMUM VALUE          0.289400E 04
          MEAN                   0.151055E 06
          STANDARD DEVIATION     0.807798E 04
FILE  1.1 SCANNED.   INITIAL    57 SAMPLES NOISE DEPENDENT.
FILE  1.1 RESET BY  210
FILE  1.1 PACKED
FILE  1.1 STATISTICS:
          (BASED ON SAMPLE SIZE OF  4800)
          MAXIMUM VALUE          0.153502E 06
          MINIMUM VALUE          0.146354E 06
          MEAN                   0.150517E 06
          STANDARD DEVIATION     0.443372E 04
FILE  1.1 PLOTTED
FILE  1.2 FIRST DIFFERENCED
FILE  1.2 STATISTICS:
          (BASED ON SAMPLE SIZE OF  4800)
          MAXIMUM VALUE           0.230000E 02
          MINIMUM VALUE          -0.240000E 02
          MEAN                   -0.874531E 00
          STANDARD DEVIATION      0.924408E 01
```

```
FILE   1.2 PLOTTED
FILE   1.3 TIDE FILTERED
FILE   1.3 STATISTICS:
           (BASED ON SAMPLE SIZE OF  4800)
           MAXIMUM VALUE          0.104613E 03
           MINIMUM VALUE         -0.125172E 03
           MEAN                  -0.349230E-01
           STANDARD DEVIATION     0.240215E 02
FILE   1.3 PLOTTED
FILE   1.4 SQUARED
FILE   1.4 STATISTICS:
           (BASED ON SAMPLE SIZE OF  4800)
           MAXIMUM VALUE          0.156681E 05
           MINIMUM VALUE          0.0
           MEAN                   0.576913E 03
           STANDARD DEVIATION     0.129861E 04
FILE   1.4 PLOTTED
FILE   1.5 INTEGRATED.  FTC =  0.0167
FILE   1.5 STATISTICS:
           (BASED ON SAMPLE SIZE OF  4800)
           MAXIMUM VALUE          0.194251E 06
           MINIMUM VALUE          0.0
           MEAN                   0.337821E 05
           STANDARD DEVIATION     0.367765E 05
```

```
FILE  1.5 PLOTTED
FILE  1.6 - FOURIER TRANSFORM
FILE  1.6 IN FFT.      2400 TERMS
          FACTORS:
              4    5    2    3    5    4    0    0    0    0    0    0    0
FILE  1.6 - POWER SPECTRUM
FILE  1.6 STATISTICS:
          (BASED ON SAMPLE SIZE OF  2400)
          MAXIMUM VALUE          0.536683E 02
          MINIMUM VALUE          0.167696E-06
          MEAN                   0.480947E 00
          STANDARD DEVIATION     0.228523E 01
FILE  1.6 PLOTTED
FILE  1.7 - FOURIER TRANSFORM
FILE  1.7 IN FFT.      2400 TERMS
          FACTORS:
              4    5    2    3    5    4    0    0    0    0    0    0    0
FILE  1.7 - POWER SPECTRUM
FILE  1.7 STATISTICS:
          (BASED ON SAMPLE SIZE OF  2400)
          MAXIMUM VALUE          0.456679E 10
          MINIMUM VALUE          0.471905E 00
          MEAN                   0.302887E 07
          STANDARD DEVIATION     0.953146E 08
FILE  1.7 PLOTTED
FINIS.
```

```
C      MAIN PROGRAM - UNUSUAL EVENT DETECTOR
C
C   ARRAYS C1 THRU C6 HOLD UP TO 40 CHARACTERS EACH FOR PLOT LABELS.
      REAL C1(10), C2(10), C3(10), C4(10), C5(10), C6(10)
C   WORKING ARRAY DECLARATIONS
      REAL Y(5000), DIFF(5000), TIDE(5000), YISE(5000)
C          Y    - RAW PRESSURE DATA
C          DIFF - FIRST DIFFERENCED DATA
C          TIDE - TIDE FILTERED DATA
C          YISE - UNUSUAL EVENT DETECTOR OUTPUT
      EQUIVALENCE (Y(1), DIFF(1), YISE(1))
C          THIS EQUIVALENCE STATEMENT CONSERVES STORAGE
C
    1 FORMAT('OFINIS.')
    2 FORMAT(10A4)
C
C   READ IN LABELS FOR PLOTS AND ASSIGN NUMBER OF FILES
C
      READ(5,2) C1, C2, C3, C4, C5, C6
      NFILES=1
C
C
      DO 100 K=1,NFILES
      FILE=FLOAT(K)+0.1
C     SET L EQUAL TO THE NUMBER OF DATA ELEMENTS TO BE READ
      L=1430
C     SET LMAX TO THE MAXIMUM SIZE OF ARRAY Y
      LMAX=5000
C
      CALL CARDS   (FILE,Y,L)
C
C   INITIAL SCAN OF DATA
C
      CALL STATS   (FILE,Y,L,YMAX,YMIN,YMEAN,YSTDEV)
      CALL SCAN    (FILE,Y,L,YMEAN,YSTDEV,L2)
C     RESET THE DATA BY 210 MINUTES
      L2=210
      CALL RESET   (FILE,Y,L,Y,L2)
C     PACK THE INTERVAL TO 1 MINUTE SAMPLES
      CALL PACK    (FILE,Y,L2,Y,LMAX, 4, 0.0, 103941)
C
C   SET THE INTERVAL TO 80 HOURS (4800 MINUTES)
      L=MIN0(4800,LMAX)
C
C   INITIAL PLOT OF DATA
C
      CALL XYAXIS  (5.,1.)
      CALL DRAW1   (FILE,Y,L,0.,0.,0,C1,C6)
```

```
C
C   FIRST DIFFERENCING
C
      THRESH=25.
      CALL FIRSTD  (FILE,Y,DIFF,L,THRESH)
      CALL DRAW1   (FILE,DIFF,L,0.,0.,0,C2,C6)
C
C   TIDE FILTER
C
      CALL ANTIDE  (FILE,DIFF,TIDE,L)
      CALL DRAW1   (FILE,TIDE,L,0.,0.,0,C3,C6)
C
C   SQUARE TIDE FILTER OUTPUT
C
      IRS=2
      CALL RECSQR  (FILE,TIDE,YISE,L,IRS)
      CALL DRAW1   (FILE,YISE,L,0.,0.,0,C3,C6)
C
C   INTEGRATE TIDE FILTER OUTPUT
C
      FTC=1./60.
      CALL INTGRL  (FILE,YISE,YISE,L,FTC)
      CALL DRAW1   (FILE,YISE,L,0.,0.,0,C4,C6)
C
C   POWER SPECTRUM OF TIDE FILTER AND DETECTOR OUTPUT
C
      CALL XYAXIS  (8.,4.)
      N=L/2
      CALL POWER   (FILE,TIDE,L)
      CALL DRAW1   (FILE,TIDE,N,0.,0.,1,C5,C6)
      CALL POWER   (FILE,YISE,L)
      CALL DRAW1   (FILE,YISE,N,0.,0.,1,C5,C6)
C
  100 CONTINUE
      WRITE(6,1)
      STOP
      END
```

BIBLIOGRAPHY

1. Titus, H. A., and Terman, F. W., "Detection of Non-stationary Processes," Proceedings of the Fourth Hawaii International Conference on System Sciences, p. 380-382, 1971.

2. *Ibid*.

3. Singleton, R. C., "An Algorithm for Computing the Mixed Radix Fast Fourier Transform," Transactions on Audio and Electroacoustics, V. AU-17, No. 2, p. 93-103, June 1969.

4. Jenkins, G. W., and Watts, D. G., Spectral Analysis and its Applications, Holden-Day, San Francisco, California, p. 209-255.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Assoc Professor Harold A. Titus, Code 52Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	ENS John F. Arfman, Jr., USN 7608 Harwood Avenue, N.E. Albuqueque, New Mexico 87110	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3 REPORT TITLE

A SOFTWARE PACKAGE FOR ANALYSIS OF GEOPHYSICAL MEASUREMENTS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; June 1971

5 AUTHOR(S) (First name, middle initial, last name)

John F. Arfman, Jr.

6. REPORT DATE

June 1971

7a. TOTAL NO. OF PAGES

79

7b. NO. OF REFS

4

8a. CONTRACT OR GRANT NO

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

A software package for general analysis of real time series for geophysical data was developed. The package consists of FORTRAN IV callable subprograms that are assigned different tasks such as input, output (either printed or graphical), initial data analysis and filtering, spurious data rejection, FFT, and spectral analysis. A geophysical problem involving detection of unusual deep ocean pressures was analyzed.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Spurious Data Rejection						
Tsunami						
Real Time Series						
Spectral Analysis						
Nyquist Frequency						

128334

Thesis
A653
c.'

Arfman
A software package for analysis of geophysical measurements.